dData

**School of Computer Science**

**B. Tech (Computer Science & Engineering) with Specialization**

**Programme Handbook**

**AcademicYear:2025-2029**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

UPES Campus, “Energy Acres” & “Knowledge Acres” Tel: + 91-135-2776053/54

Bidholi & Kandoli Fax: + 91-135-2776090

Dehradun, Uttarakhand-248001 URL: [www.upes.ac.in](http://www.upes.ac.in)

**Curriculum development and Approval Process**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Release Date | Version No: | Approved in AC | Released by School | Verified by IQAC |
| **25th April 2024** | 1 | **25th April 2024** | **YES** |  |
| Event | Presented By | Approval Date | Signature | Office Of |
| CCC | Cluster Heads | 5th February 2025 |  | Dean SOCS |
| BOS | BOS Committee | 5th March 2025 |  | Dean SOCS |
| Academic Council | Academic Council Committee | **25th April 2025** |  | Dean SoCS |

**Contents**

[Intellectual Property Rights 0](file:///C:\Users\p.priya\Desktop\OneDrive_2023-09-28\04%20Programme%20Handbook%20Templates\School%20of%20Computer%20Science\UPES_IQAC_PROGRAMME%20HANDBOOK_B.Tech%20CSE.docx#_Toc149495638)

[1.0 Abbreviations 1](#_Toc149495639)

[1.0 Director/ Dean Message 1](#_Toc149495640)

[2.0 Vision and Mission of the University: 2](#_Toc149495641)

[3.0 Vision and Mission of the School 2](#_Toc149495642)

[4.0 About the Cluster/ Department/ Centre 3](#_Toc149495643)

[5.0 Programme Overview 4](#_Toc149495644)

[6.0 Programme Educational Objectives 4](#_Toc149495645)

[7.0 Programme Outcome and Programme Specific Outcomes 4](#_Toc149495646)

[8.0 Academic Integrity Policy](#_Toc149495647) …………………………………………………………...6

[9.0 Overview of Credit Allocation/ Credit Break up 6](#_Toc149495648)

[10.0 Programme Structure 8](#_Toc149495649)

[11.0 List of Electives 14](#_Toc149495650)

[12.0 Course Syllabus/ Course Plans 15](#_Toc149495651)

# Intellectual Property Rights

All Information contained in this document has been licensed to the UPES, which have the sole intellectual property rights in this information. By accepting this material, the recipient agrees that the information contained herein will be held in confidence and will not be reproduced, disclosed, divulged, or used either in whole or in part without prior permission from UPES @ UPES

# Abbreviations

|  |  |  |
| --- | --- | --- |
| Cat | - | Category |
| Cr | - | Credits *(A credit is equivalent to one lecture hour/ one hour of tutorial/ two hours of Laboratory)* |
| L | - | Lecture |
| T | - | Tutorial |
| P | - | Practical |
| ENGG | - | Engineering Sciences (including General, Core) |
| HUM | - | Humanities (including Languages, Social Sciences, and others) |
| SCI | - | Basic Sciences (including Mathematics) |
| PRJ | - | Project Work (including Seminars, Dissertation, and Internships) |
| PE | - | Program Elective (includes Specialization courses) |
| TC | - | Total Credits |
|  |  |  |
| AIE | - | Computer Science and Engineering-Artificial Intelligence |
| CCE | - | Computer Science and Communication Engineering |
| CHY | - | Chemistry |
| CSE | - | Computer Science and Engineering |
| CVL | - | Civil Engineering |
| CUL | - | Cultural Education |
| MAT | - | Mathematics |
| PHY | - | Physics |
| UE | - | University Elective (includes Signatory, Exploratory and Open Electives) |

# Dean’s Message

Welcome to UPES-School of Computer Science's B. Tech in Computer Science and Engineering program. Our curriculum combines theory and hands-on experience to prepare you for the fast-changing world of Computer Science. Our faculty consists of accomplished researchers and industry professionals who are committed to academic excellence. In addition to traditional classrooms, you will have opportunities for real-world applications, networking, and industry partnerships. Embrace challenges, approach your academic journey with enthusiasm, and remember that you are part of a community dedicated to knowledge and innovation. With your dedication and the resources provided by UPES-SoCS, you will become a skilled professional and visionary leader to make a transformative impact on the global stage.

# Vision and Mission of the University:

**Vision of UPES**

To be an Institution of Global standing for developing professionally competent talent contributing to nation building.

**Mission of UPES**

* Develop industry-focused professionals with an international outlook.
* Foster effective outcome-based education system to continually improve teaching-learning and research.
* Inculcate integrative thought process among students to instill lifelong learning.
* Create global knowledge eco-system through training, research & development and consultancy.
* Practice and promote high standards of professional ethics and develop harmonious relationship with environment and society.

# Vision and Mission of the School

**Vision**

To create a dynamic, collaborative, and inclusive community that advances technology, fosters creativity, and empowers individuals to make a positive global impact. The school aspires to lead and inspire in the development of new technology, and the application of technology to solve problems of society.

**Mission**

Our mission is to empower the next generation of professionals in Computer Science with the knowledge, skills, and innovative mindset required to shape the future. We are dedicated to fostering a vibrant learning community where students, faculty, and industry partners collaborate to advance society and the frontiers of technology.

* + Excellence in Education: Delivering a rigorous and multi-disciplinary computer science curriculum that equips our students with a deep understanding of foundational principles and the ability to adapt to evolving technologies.
  + Research and Innovation: Striving to be at the forefront of technology research, pushing boundaries in areas such as artificial intelligence, data science, IoT and cybersecurity.
  + Industry Engagement: Actively engaging with industry partners to ensure that our programs remain relevant and provide opportunities for students to apply their knowledge in real-world settings. Encouraging our faculty members to collaborate with industry partners on real-world projects.
  + Entrepreneurship: Strongly encouraging entrepreneurship both among the faculty members and the students.
  + Ethical Computing: Emphasizing the importance of ethical and responsible use of technology, instilling in our faculty, students, and the society at large a sense of social responsibility and ethical decision-making.
  + Lifelong Learning: Encouraging a culture of lifelong learning among our students, faculty, and alumni, ensuring that they continue to adapt and thrive in the ever-changing field of computer science.

Through these commitments, we aim to produce graduates who are not only technically proficient but also visionary leaders who will drive positive change in the world through the transformative power of computer science.

# About the School

The UPES School of Computer Science is founded with futuristic approach, to provide cutting-edge computer science education, to empower future generations to harness the transformative power of technology. The school is organized into four clusters named Artificial Intelligence, Data Science, Cloud & S/W Operations and Systems. The SoCS faculty comprises accomplished researchers and industry professionals dedicated to academic excellence. The school offers various programs like B.Tech., BCA, B.Sc., M.Tech., MCA, Ph.D. A well-crafted curriculum aligned with the industry needs, well-equipped laboratories, innovative pedagogy, are the strength of the school.

**Research Focus**

The school research focus encompasses cutting-edge domains, including artificial intelligence, data science, IoT, and cybersecurity. We delve into innovative studies at the intersection of these fields, driving advancements in technology.

# Programme Overview

The Bachelor of Technology in Computer Science and Engineering program provides a comprehensive education in the fundamental principles and advanced concepts of computer science. Students engage in a rigorous curriculum covering areas such as algorithms, data structures, software engineering, artificial intelligence, and machine learning. The program offers various specializations through it’s program elective courses. The program emphasizes hands-on experience through practical projects, internships, and industry collaborations, enabling students to apply theoretical knowledge to real-world scenarios. With a focus on innovation and problem-solving, B.Tech CSE equips graduates with the skills to excel in diverse roles within the rapidly evolving field of computer science, fostering a strong foundation for professional success and contributions to technological advancements

# Programme Educational Objectives

**Technical Excellence:**

Graduates will demonstrate a strong foundation in Computer Science and its applications. They will excel in technical roles in software industries, including design, development, problem-solving, and production support.

**Continuous Learning and Advancement:**

Graduates will pursue higher education in reputable institutions, enriching their knowledge and specialisation in Computer Science and Applications.

**Innovation and Adaptability:**

Graduates will innovate and adapt evolving technologies to address contemporary issues in various domains of computer applications creatively.

**Ethical and Social Responsibility:**

Graduates will exhibit ethical and social responsibility as solution providers and entrepreneurs in information technology. They will uphold high ethical values and demonstrate empathy for societal needs.

# Programme Outcome and Programme Specific Outcomes

**Programme Outcomes**

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

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

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

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

10. 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.

11. 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.

12. 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.

**Programme Specific Outcomes**

1. Perform system and application programming using computer system concepts, the principles of data structures, algorithm development, problem-solving, and optimization techniques.
2. Apply software development and project management methodologies, integrating principles from both front-end and back-end development, and effectively utilize contemporary tools and technologies.
3. Exhibit a commitment to ethical practices, societal responsibilities, and continuous learning, contributing to the advancement of technology and addressing challenges in diverse computing domains.

# Academic Integrity Policy

# For Academic Integrity Policy, refer Academic Handbook.

# Overview of Credit Allocation/ Credit Break up

Category-wise Credit distribution

|  |  |
| --- | --- |
| **Category** | **Number of Credits** |
| Major Core (MC) | 60 |
| Basic Sciences - Core (SCI) | 26 |
| Engineering Sciences - Core (ENGG) | 08 |
| Major Elective (ME/PE) | 24 |
| Life Skill Courses (LSC)\* | 12 |
| Exploratory Courses (EC)\* | 18 |
| Projects (PRJ) | 16 |
| **Total** | **164** |

\* Electives

* Major core subjects include those subjects that are mandatory to all similar programmes and program specific courses. To be eligible for the degree, students must successfully finish each of the courses.
* Major elective courses provide the students the opportunity to study courses that are more complex and specialized, in their field of specialization.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Major Core (MC) Total number of Credits: 63 Credits** | | | | | |
| **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
| CSEG1126 | Linux Lab | 0 | 0 | 2 | 2 |
| CSEG1041 | Programming in C | 3 | 0 | 2 | 5 |
| CSEG1043 | Data Structures and algorithms | 4 | 0 | 1 | 5 |
| CSEG1021 | Python Programming | 2 | 0 | 2 | 4 |
| CSEG2072 | Database Management Systems | 3 | 0 | 2 | 5 |
| CSEG2006 | Discrete Mathematical Structures | 3 | 0 | 0 | 3 |
| CSEG1044 | Object Oriented Programming | 3 | 0 | 1 | 4 |
| CSEG2060 | Operating Systems | 3 | 0 | 0 | 3 |
| CSEG2064 | Software Engineering | 3 | 0 | 0 | 3 |
| CSAI2018 | Elements of AIML | 2 | 0 | 1 | 3 |
| CSEG2065 | Data communication and Networks | 3 | 0 | 1 | 4 |
| CSEG3053 | Design and Analysis of Algorithms | 3 | 0 | 1 | 4 |
| CSEG3040 | Cryptography and Network Security | 3 | 0 | 0 | 3 |
| CSEG3055 | Formal Languages and Automata Theory | 3 | 0 | 0 | 3 |
| CSEG3002 | Object Oriented Analysis and Design | 3 | 0 | 0 | 3 |
| CSEG3015 | Compiler Design | 3 | 0 | 0 | 3 |
| CSEG4038 | IT Ethical Practices | 3 | 0 | 0 | 3 |
| **Total Credits** | | | | | **60** |
| **BASIC SCIENCES-Core (SCI) Total Number of Credits: 26 Credits** | | | | | |
| **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
| MATH1059 | Advanced Engineering Mathematics – 1 | 3 | 1 | 0 | 4 |
| SSEN0101 | Environmental Sustainability and Climate Change | 2 | 0 | 0 | 2 |
| PHYS1036 | Physics for Computer Engineers | 4 | 0 | 1 | 5 |
| MATH1065 | Advanced Engineering Mathematics – 2 | 3 | 1 | 0 | 4 |
| SSEN0102 | Environmental Sustainability and Climate Change (Living Lab) | 0 | 0 | 2 | 2 |
| MATH2059 | Linear Algebra | 3 | 0 | 0 | 3 |
| CSEG3056 | Probability, Entropy, and MC Simulation | 3 | 0 | 0 | 3 |
| CSEG3057 | Statistics and Data Analysis | 3 | 0 | 0 | 3 |
|  | **Total Credits** |  |  |  | **26** |
| **Engineering Sciences Total Number of Credits: 8 Credits** | | | | | |
| **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
| CSEG1027 | Problem Solving | 2 | 0 | 0 | 2 |
| CSEG1032 | Computer organization and Architecture | 3 | 0 | 0 | 3 |
| ECEG1012 | Digital Electronics | 3 | 0 | 0 | 3 |
| **Total Credits** | | | | | **8** |
| **Major Elective (ME) Total Number of Credits: 24 Credits** | | | | | |
| **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
|  | PE-1 | 4 | 0 | 1 | 5 |
|  | PE-2 | 4 | 0 | 1 | 5 |
|  | PE-3 | 4 | 0 | 1 | 5 |
|  | PE-4 | 4 | 0 | 1 | 5 |
|  | PE-5 | 3 | 0 | 1 | 4 |
| **Total Credits** | | | | | **24** |
| **Life Skills Courses Total Number of Credits: 10 Credits** | | | | | |
| **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
| SFLS0001 | Managing Self |  |  |  | 2 |
| SFLS0002 | Time & Priority Management |  |  |  | 2 |
| SFLS0003 | Communication Skills (Oral) |  |  |  | 2 |
| EMPL002 | EDGE-SoftSkills | 1 | 0 | 0 | 0 |
| SFLS0004 | Communication Skills (Written) |  |  |  | 2 |
| EMPL003 | EDGE – Advance Communication | 1 | 0 | 0 | 0 |
| SFLS0005 | Working with People |  |  |  | 2 |
| SFLS0006 | Deepening Self |  |  |  | 2 |
| **Total Credits** | | | | | **12** |
| **Exploratory Courses Total Number of Credits: 15 Credits** | | | | | |
| **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
|  | Exploratory-1 | 3 | 0 | 0 | 3 |
|  | Exploratory-2 | 3 | 0 | 0 | 3 |
|  | Exploratory-3 | 3 | 0 | 0 | 3 |
|  | Exploratory-4 | 3 | 0 | 0 | 3 |
|  | Exploratory-5 | 3 | 0 | 0 | 3 |
|  | Exploratory-6 | 3 | 0 | 0 | 3 |
| **Total Credits** | | | | | **18** |
| **Projects (PRJ) Total Number of Credits: 16 Credits** | | | | | |
| **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
| PROJ3154 | Minor Project | 0 | 0 | 5 | 5 |
| PROJ4145 | Capstone Project - Phase-1 | 0 | 0 | 5 | 5 |
| SIIB4102 | Summer Internship | 0 | 0 | 0 | 1 |
| SLLS2001 | Social Internship | 0 | 0 | 0 | 0 |
| PROJ4146 | Capstone Project - Phase-2 | 0 | 0 | 5 | 5 |
| **Total Credits** | | | | | **16** |

# Programme Structure

The term "Program Structure" refers to a list of courses (Core, Elective, and Open Elective) that make up an academic program, describing the syllabus, credits, hours of instruction, assessment and examination systems, minimum amount of credits necessary for program graduation, etc.

**B. Tech (Computer Science & Engineering) with Specialization**

Semester I:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **prerequisites** |
| SCI | MATH1059 | Advanced Engineering Mathematics – 1 | 3 | 1 | 0 | 4 |  |
| MC | CSEG1041 | Programming in C | 3 | 0 | 2 | 5 |  |
| MC | CSEG1126 | Linux Lab | 0 | 0 | 2 | 2 |  |
| ES | CSEG1027 | Problem Solving | 2 | 0 | 0 | 2 |  |
| SCI | SSEN0101 | Environment Sustainability & Climate Change | 2 | 0 | 0 | 2 |  |
| SCI | PHYS1036 | Physics for Computer Engineers | 4 | 0 | 1 | 5 |  |
| LSC | SFLS0001 | Managing Self |  |  |  | 2 |  |
| **Semester Credits** | | | | | | | **22** |

Semester II:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **Prerequisites** |
| MC | CSEG1043 | Data Structures and Algorithms | 4 | 0 | 1 | 5 | Programming in C |
| MC | CSAI2018 | Elements of AIML | 2 | 0 | 1 | 3 | Python Programming |
| MC | CSEG1021 | Python Programming | 2 | 0 | 2 | 4 |  |
| ES | ECEG1012 | Digital Electronics | 3 | 0 | 0 | 3 |  |
| SCI | MATH1065 | Advanced Engineering Mathematics – 2 | 3 | 1 | 0 | 4 | Advanced Engineering Mathematics - I |
| SCI | SSEN0102 | Environment Sustainability and Climate Change (living Lab) | 0 | 0 | 2 | 2 |  |
| LSC | SFLS0002 | Time & Priority Management |  |  |  | 2 |  |
| **Semester Credits** | | | | | | | 23 |

Semester III:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **Prerequisites** |
|  |  |  |  |  |  |  |  |
| MC | CSEG2060 | Operating Systems | 3 | 0 | 0 | 3 | Linux Lab |
| MC | CSEG2065 | Data communication and Networks | 3 | 0 | 1 | 4 |  |
| ES | CSEG1032 | Computer Organization and Architecture | 3 | 0 | 0 | 3 |  |
| MC | CSEG3053 | Design and Analysis of Algorithms | 3 | 0 | 1 | 4 | Data structures and algorithms |
| LSC | SLLS2001 | Social Internship | 0 | 0 | 0 | 0 |  |
| EC |  | Exploratory-1 | 3 | 0 | 0 | 3 |  |
| LSC | SFLS0003 | Communication Skills (Oral) |  |  |  | 2 |  |
| ME |  | PE-1 | 4 | 0 | 1 | 5 |  |
| **Semester Credits** | | | | | | | 24 |

Semester IV:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **Prerequisites** |
| MC | CSEG1044 | Object Oriented Programming | 3 | 0 | 1 | 4 | Programming in C |
| MC | CSEG2072 | Database Management Systems | 3 | 0 | 2 | 5 |  |
| MC | CSEG2006 | Discrete Mathematical Structures | 3 | 0 | 0 | 3 |  |
| SCI | MATH2059 | Linear Algebra | 3 | 0 | 0 | 3 |  |
| LSC | SLLS2004 | Indian Constitution | 0 | 0 | 0 | 0 |  |
| LSC | EMPL002 | EDGE-SoftSkills | 1 | 0 | 0 | 0 |  |
| EC |  | Exploratory-2 | 3 | 0 | 0 | 3 |  |
| LSC | SFLS0004 | Communication Skills (Written) |  |  |  | 2 |  |
| ME |  | PE-2 | 4 | 0 | 1 | 5 |  |
| **Semester Credits** | | | | | | | 25 |

Semester V:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **Prerequisites** |
| MC | CSEG3040 | Cryptography and Network Security | 3 | 0 | 0 | 3 | Data Communication and Networks |
| MC | CSEG3055 | Formal Languages and Automata Theory | 3 | 0 | 0 | 3 | Discrete Mathematical Structures |
| MC | CSEG3002 | Object Oriented Analysis and Design | 3 | 0 | 0 | 3 | Software Engineering |
| SCI | CSEG3056 | Probability, Entropy, and MC Simulation | 3 | 0 | 0 | 3 |  |
| EC |  | Exploratory-3 | 3 | 0 | 0 | 3 |  |
| LSC | SFLS0005 | Working with People |  |  |  | 2 |  |
| LSC | EMPL003 | EDGE – Advance Communication |  |  |  | 0 |  |
| ME |  | PE-3 | 4 | 0 | 1 | 5 |  |
| **Semester Credits** | | | | | | | 22 |

Semester VI:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **Prerequisites** |
| SCI | CSEG3057 | Statistics and Data Analysis | 3 | 0 | 0 | 3 |  |
| MC | CSEG2064 | Software Engineering | 3 | 0 | 0 | 3 |  |
| MC | CSEG3015 | Compiler Design | 3 | 0 | 0 | 3 | Formal Languages and Automata Theory |
| LSC | EMPL004 | EDGE – Advance Communication II | 1 | 0 | 0 | 0 |  |
| LSC | SFLS0006 | Deepening Self |  |  |  | 2 |  |
| PRJ | PROJ3154 | Minor Project | 0 | 0 | 5 | 5 |  |
| EC |  | Exploratory-4 | 3 | 0 | 0 | 3 |  |
| ME |  | PE-4 | 4 | 0 | 1 | 5 |  |
| **Semester Credits** | | | | | | | 24 |

Semester VII:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **Prerequisites** |
| PRJ | PROJ4145 | Capstone Project - Phase-1 | 0 | 0 | 5 | 5 |  |
| PRJ | SIIB4102 | Summer Internship | 0 | 0 | 0 | 1 |  |
| EC |  | Exploratory-5 | 3 | 0 | 0 | 3 |  |
| ME |  | PE-5 | 3 | 0 | 1 | 4 |  |
| **Credits Semester** | | | | | | | 13 |

Semester VIII:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cat** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** | **Prerequisites** |
| PRJ | PROJ4146 | Capstone Project - Phase-2 | 0 | 0 | 5 | 5 | Capstone Project - Phase-1 |
| EC |  | Exploratory -6 | 3 | 0 | 0 | 3 |  |
| MC | CSEG4038 | IT Ethical Practices | 3 | 0 | 0 | 3 |  |
| **Credits Semester** | | | | | | | 11 |

**Specialization Tracks**

The students enrolled in B.Tech Computer Science would have an option to specialize in one the following emerging areas:

1. Artificial Intelligence and Machine Learning
2. DevOps
3. Cloud Computing And Virtualization Technology
4. Full Stack Development
5. Cyber Security and Digital Forensics
6. Big Data
7. Data Science
8. Internet of Things (IoT)
9. Graphics & Gaming
10. Core Computer Science

The student must complete a minimum of 24 credits in the chosen area of specialization.

List of elective courses in specialization tracks

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Program Elective 24 Credits** | | | | | | |
| **Track 1 : Artificial Intelligence and Machine Learning** | | | | | | |
| **PE** | **Course Code** | **Course Title** | **L** | **T** | **P** | **TC** |
| PE-1 | CSAI2017P | Applied Machine Learning | 4 | 0 | 1 | 5 |
| PE-2 | CSAI3027P | Deep Learning | 4 | 0 | 1 | 5 |
| PE-3 | CSAI3028P | Pattern and Visual Recognition | 4 | 0 | 1 | 5 |
| PE-4 | CSEG4040P | Computational Linguistics and Natural Language Processing | 4 | 0 | 1 | 5 |
| PE-5 | CSDS4004P | Generative Artificial Intelligence | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 2 : DevOps** | | | | | | |
| PE-1 | CSDV2010P | DevOps Fundamentals and SCM | 4 | 0 | 1 | 5 |
| PE-2 | CSDV3024P | DevSecOps: Integrating security into DevOps practices | 4 | 0 | 1 | 5 |
| PE-3 | CSDV3023P | Container Orchestration and Security | 4 | 0 | 1 | 5 |
| PE-4 | CSDV4014P | CICD Pipeline and Security | 4 | 0 | 1 | 5 |
| PE-5 | CSDV4013P | System Provisioning and Monitoring | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 3 : Cloud Computing And Virtualization Technology** | | | | | | |
| PE-1 | CSVT2011P | Cloud Computing Fundamentals | 4 | 0 | 1 | 5 |
| PE-2 | CSVT3032P | Cloud Computing Architecture and Deployment Models | 4 | 0 | 1 | 5 |
| PE-3 | CSDV4016P | Containerization and DevOps | 4 | 0 | 1 | 5 |
| PE-4 | CSVT4022P | Cloud Application Development | 4 | 0 | 1 | 5 |
| PE-5 | CSVT4024P | Cloud Computing Security and Management | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 4 : Full Stack Development** | | | | | | |
| PE-1 | CSFS2004 | Frontend Development | 4 | 0 | 1 | 5 |
| PE-2 | CSFS3008P | Backend Development | 4 | 0 | 1 | 5 |
| PE-3 | CSFS3009P | Microservices and Spring-Boot | 4 | 0 | 1 | 5 |
| PE-4 | CSVT4023P | Cloud Computing and Security | 4 | 0 | 1 | 5 |
| PE-5 | CSDV4015P | Container Orchestration and Security | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 5 : Cyber Security and Digital Forensics** | | | | | | |
| PE-1 | CSSF2015P | Information Technology and Cyber Security | 4 | 0 | 1 | 5 |
| PE-2 | CSSF3030P | Ethical Hacking & Penetration Testing | 4 | 0 | 1 | 5 |
| PE-3 |  | Network Security Practices | 4 | 0 | 1 | 5 |
| PE-4 | CSSF4021P | Digital Forensics | 4 | 0 | 1 | 5 |
| PE-5 | CSSF4023P | OS, Application & Cloud Security | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 6 : Big Data** | | | | | | |
| PE-1 | CSBD2011P | Big Data Overview and Ingestion | 4 | 0 | 1 | 5 |
| PE-2 | CSBD3020P | Big Data Storage and Analysis | 4 | 0 | 1 | 5 |
| PE-3 | CSBD3019P | Big Data Processing – Disk based and In Memory | 4 | 0 | 1 | 5 |
| PE-4 | CSBD4012 | Stream Processing | 4 | 0 | 1 | 5 |
| PE-5 | CSBD4011P | Big Data Search and Security | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 7 : Data Science** | | | | | | |
| PE-1 | CSDS2002P | Fundamentals of Data Science | 4 | 0 | 1 | 5 |
| PE-2 | CSDS3005P | Data Visualization and Interpretation | 4 | 0 | 1 | 5 |
| PE-3 | CSDS3006P | Machine Learning and Deep Learning | 4 | 0 | 1 | 5 |
| PE-4 | CSEG4040P | Computational Linguistic and Natural Language Processing | 4 | 0 | 1 | 5 |
| PE-5 | CSDS4004P | Generative Artificial Intelligence | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 8: Internet of Things (IoT)** | | | | | | |
| PE-1 | CSIS2013P | Introduction to IoT, Sensors and Microcontrollers | 4 | 0 | 1 | 5 |
| PE-2 | CSIS3024P | IoT Network Architecture and Communication Protocols | 4 | 0 | 1 | 5 |
| PE-3 | CSIS3023P | Industrial IoT and ARM based Embedded Programming | 4 | 0 | 1 | 5 |
| PE-4 | CSIS4015P | Single Board Computers and IoT Applications Development | 4 | 0 | 1 | 5 |
| PE-5 | CSBA4023P | Data Analytics for IoT | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |
| **Track 9: Graphics & Gaming** | | | | | | |
| PE-1 |  | Interactive Design and 3D Animation | 4 | 0 | 1 | 5 |
| PE-2 | CSGG3024P | Game Programming | 4 | 0 | 1 | 5 |
| PE-3 | CSGG3023P | Computer Graphics | 4 | 0 | 1 | 5 |
| PE-4 | CSGG4015P | Augmented and Virtual Reality Development | 4 | 0 | 1 | 5 |
| PE-5 | CSGG4016P | Web Programming for Interactive 3D Graphics | 3 | 0 | 1 | 4 |
| **Track 10: Core Computer Science** | | | | | | |
| PE-1 |  | Applications of Machine Learning in Industry | 4 | 0 | 1 | 5 |
| PE-1 | CSFS2004 | \*Frontend Development | 4 | 0 | 1 | 5 |
| PE-1 | CSGG3023P | \*Computer Graphics | 4 | 0 | 1 | 5 |
| PE-1 |  | Information Theory and coding | 4 | 0 | 1 | 5 |
| PE-2 | CSDS2002P | \*Fundamentals of Data Science | 4 | 0 | 1 | 5 |
| PE-2 | CSFS3008P | \*Backend Development | 4 | 0 | 1 | 5 |
| PE-2 |  | Signal Processing | 4 | 0 | 1 | 5 |
| PE-2 |  | Real Time Operating Systems | 4 | 0 | 1 | 5 |
| PE-3 |  | Cloud Computing Architecture and Design | 4 | 0 | 1 | 5 |
| PE-3 |  | Graph Theory | 4 | 0 | 1 | 5 |
| PE-3 |  | Digital Image Processing | 4 | 0 | 1 | 5 |
| PE-3 |  | Microcontroller and Embedded Systems | 4 | 0 | 1 | 5 |
| PE-4 |  | Big Data Overview | 4 | 0 | 1 | 5 |
| PE-4 |  | Wireless Sensor Network | 4 | 0 | 1 | 5 |
| PE-4 |  | Operation Research & Game Theory | 4 | 0 | 1 | 5 |
| PE-4 |  | Parallel Computing | 4 | 0 | 1 | 5 |
| PE-5 |  | Devops Overview | 3 | 0 | 1 | 4 |
| PE-5 |  | Distributed Systems | 3 | 0 | 1 | 4 |
| PE-5 |  | Mobile Computing | 3 | 0 | 1 | 4 |
| PE-5 |  | Fog & Edge Computing | 3 | 0 | 1 | 4 |
|  |  |  |  |  |  |  |

\*Already running as PE in other specializations

# Course Syllabus/ Course Plans

**Sample Course plan template-**

**SEMESTER I**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **MATH1059** | **Advanced Engineering Mathematics – 1** | | 3 | 1 | 0 | 4 |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 60** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The course aims to

1. Provide both theoretical as well as practical use of differential and integral calculus.
2. Employ vector analysis and vector calculus for modeling physical and engineering problems.
3. Develop capability and skill set to model situations governed by linear and nonlinear differential equations.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Understand the power of differential calculus for gaining insight into problems of interest.

**CO2.** Recognise the potiential of integral calculas to gain understanding of interesting real world challenges

**CO3.** Explore the concept of vector-valued functions as applied in engineering applications.

**CO4.** Formulate and analyze mathematical models of a variety of real-world problems.

**CO5.** Develop and visualize solutions of nonlinear mathematical models.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 3 | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | 3 | 3 | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | 3 | 3 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 4** | 3 | 3 | 3 | **-** | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 5** | 3 | 3 | 3 | **-** | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **Average** | 3 | 3 | 2 | **-** | 1.8 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit 0: Motivation** **1 Lecture Hours**

Why study this course? Application of Calculus and Differential equations in addressing real-world problems.

**Unit I: Differential Calculus 12 Lecture Hours**

Review: Functions and their graphs, polynomial, exponential, and logarithmic functions, Remainder and Factor theorems of polynomials.Limits, Continuity and Differentiability, Rolle’s theorem, Lagrange’s and Cauchy mean value theorems, Successive differentiation, Leibnitz’s theorem, Taylor’s series without proof, Lagrange’s form of remainder, Functions of several variables, Partial differentiation, Euler’s theorem, Jacobian, Maxima and minima. Recap of Unit-I.

**Unit II: Integral Calculus 12 Lecture Hours**

Definite integrals and properties, Double integrals: Cartesian and Polar co-ordinates, Cylindrical and Spherical coordinates, Change of order of integration, Change of variables, Triple integrals, Gamma, Beta functions and their properties, Applications of Calculus: Real-world problems. Recap of Unit-II.

**Unit III: Vector Algebra and Calculus 12 Lecture Hours**

Motivation, Vector algebra, Scalar and vector point functions, Vector Differentiation, Gradient, Divergence and Curl, Vector Integration: Line integral, Surface integral, Volume integral, Application of integrals: Work, Circulation and Flux, Green, Gauss, and Stokes theorems (without proof) and their applications. Recap of Unit-III.

**Unit IV: Linear Differential Equations with Visualizations 13 Lecture Hours**

Motivation: Linearity vs non-linearity, Linear superposition principle, First-order linear differential equations: Exact differential equations. Integrating Factors. Applications of first-order linear equations: Electric circuits, Radioactive decay, Population growth. Second and Higher order linear differential equations: Solution of homogeneous and non-homogeneous equations with constant coefficients, Wronskian, Solution of second-order differential equation by variation of parameters, Autonomous vs Non-autonomous systems, Applications: Harmonic oscillator, Electric Circuits. Visualization Tool: GeoGebra https://www.geogebra.org/t/differential-equation. Recap of Unit-IV.

**Unit V: Non-Linear Differential Equations with Visualizations 10 Lecture Hours**

Brief history of interdisciplinary studies of nonlinear systems, Pendulum equations, Phase portraits, Linearization around equilibrium, Nonlinear models: Logistic equation, Bass model of diffusion of innovation, Lotka-Volterra predator-prey model, Epidemic models-SIR, Application to the nerve impulse. Visualization Tool: GeoGebra https://www.geogebra.org/t/differential-equation. Recap of Unit V.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. G. B. Thomas, and R. L. Finney, "Calculus and Analytical Geometry", 9th Edition, Pearson Education India, 2010. 2. 2. G. James, and P. Dyke, "Advanced Modern Engineering Mathematics", 5th Edition, Pearson Education, 2018. 3. E. Kreyszig, "Advanced Engineering Mathematics", 10th Edition, J. Wiley and Sons, 2023. |
| **Reference books** | 1. 1. T. M. Apostol, "Mathematical Analysis", 2nd Edition, Narosa, 2002. 2. 2. M. Braun, "Differential Equations and their Applications", 4th Edition, Springer, 1993. 3. S. H. Strogatz, "Nonlinear Dynamics and Chaos", 2nd Edition, CRC Press, 2014. 4. 4. R. H., Enns, "It’s a Nonlinear World", Springer, 2010. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG1041** | **Programming in C** | | **3** | **0** | **2** | **5** |
| **Total Units to be Covered: 7** | | **Total Contact Hours: 105** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Introduce students to the basic principles and concepts of programming.
2. Develop students' ability to solve programming problems by applying the fundamental concepts of C programming.
3. Demonstrate students how to design and develop structured programs using modular programming techniques.
4. Enable students to apply their C programming skills to develop small-scale applications.

**Course Outcomes**

On completion of this course, the students will be able to

1. Demonstrate a high level of proficiency in writing correct and efficient C code.
2. Acquire the skills to debug and troubleshoot C programs efficiently.
3. Expertise in designing structured programs using modular programming techniques.
4. Understand file and memory management techniques.
5. Acquire the ability to apply their C programming skills to develop practical applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | **-** | **-** | .4 | **-** | 2 | **-** | **-** | **-** | .6 | **-** | **-** | .4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Syllabus**

**Unit I: Introduction to Computing 7 Lecture Hours**

Basic computer organization, Evolution of programming languages, Data representation and storage, Basics of programming environment: editors, debuggers, translators, basics of program design and execution, Algorithms, Pseudocode and Flowcharts.

**Unit II: C Programming Fundamentals 7 Lecture Hours**

Data types and type conversion, variables (declaration vs definition, local vs global), keywords, header files, structure of a C program. Operators: types of operators (arithmetic, relational, logical, bit-wise, increment/decrement, assignment, sizeof, ternary), operator precedence and associativity. Conditional statements: if, else, switch-case, break, continue, goto, label. Loops: for, while and do-while.

**Unit III: Array and Function 9 Lecture Hours**

Array, Multi-dimensional arrays, Strings, Function, Pass and Return by value, Pass and Return by Reference, Recursion, Scope Rules.

**Unit IV: Structures and Pointers 8 Lecture Hours**

Structure, typedef, Union, Enum, Bit-Fields, Pointer, Pointer to Arrays, Pointer Arrays, Pointer to Pointers, Address Arithmetic, Pointer to Structures, Pointer to functions, Bit-wise operator.

**Unit V: File handling, Memory management 7 Lecture Hours**

Data Organization, File Operations. Dynamic Memory Management: Malloc(), Calloc(), Realloc() and Free (), Garbage Collection.

**Unit VI: Preprocessor, Macro, Static and Shared Library 7 Lecture Hours**

Preprocessor & Directives, Macro, Macro vs Functions, C standard library: stdio.h, ctype.h, stdlib.h, assert.h, stdarg.h, time.h etc., Compilation of a C Program, Static Library, Shared Library.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. B. W. Kernighan, and D. M. Ritchie, "The C programming language", 2nd Edition, Prentice Hall, 1988. 2. P. J. Deitel, and H. M. Deitel, "C: How to program", 8th Edition, Pearson Education, 2015. |
| **Reference books** | 1. B. S. S. Gottfried, "Schaum's Outline of Programming with C", 2nd Edition, McGraw-Hill, 1996. 2. P. V. D. Linden, "Expert C Programming-Deep C Secrets", Pearson Education, 1994. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Programming in C Lab**

**List of Experiments**

**Experiment 1: Installation, Environment Setup and starting with C language**

1. Write a C program to print “ Hello World”
2. Write a C Program to print the address in multiple lines (new line).
3. Write a program that prompts the user to enter their name and age.
4. Write a C program to add two numbers, take number from user.

**Experiment 2: Operators**

1. WAP a C program to calculate the area and perimeter of a rectangle based on its length and width.
2. WAP a C program to Convert temperature from Celsius to Fahrenheit using the formula: F = (C \* 9/5) + 32.

**Experiment 3.1:** Conditional Statements

1. WAP to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle, or scalene. Take sides of the triangle as input from a user.
2. WAP to compute the BMI Index of the person and print the BMI values as per the following ranges. You can use the following formula to compute BMI= weight(kgs)/Height(Mts)\*Height(Mts).

|  |  |
| --- | --- |
|  | BMI |
| Starvation | <15 |
| Anorexic | 15.1 to 17.5 |
| Underweight | 17.6 to 18.5 |
| Ideal | 18.6 to 24.9 |
| Overweight | 25 to 25.9 |
| Obese | 30 to 39.9 |
| Morbidity Obese | 40.0 above |

1. WAP to check if three points (x1,y1), (x2,y2) and (x3,y3) are collinear or not.
2. According to the gregorian calendar, it was Monday on the date 01/01/01. If Any year is input through the keyboard write a program to find out what is the day on 1st January of this year.
3. WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangles should be three.

**Experiment 3.2: Loops**

1. WAP to enter numbers till the user wants. At the end, it should display the count of positive, negative, and Zeroes entered.
2. WAP to print the multiplication table of the number entered by the user. It should be in the correct formatting.  Num \* 1 = Num
3. WAP to generate the following set of output.

1

2 3

4 5 6



1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1. The population of a town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 years. Write a program to determine the population at the end of each year in the last decade.
2. Ramanujan Number is the smallest number that can be expressed as the sum of two cubes in two different ways. WAP to print all such numbers up to a reasonable limit.

Example of Ramanujan number: 1729

12^3 + 1^3 and 10^3 + 9^3. for a number L=20(that is limit)

**Experiment 4: Variable and Scope of Variable**

1. Declare a global variable outside all functions and use it inside various functions to understand its accessibility.
2. Declare a local variable inside a function and try to access it outside the function. Compare this with accessing the global variable from within the function.
3. Declare variables within different code blocks (enclosed by curly braces) and test their accessibility within and outside those blocks.
4. Declare a static local variable inside a function. Observe how its value persists across function calls.

**Experiment 5: Array**

1. WAP to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers.
2. WAP to read a list of integers and store it in a single dimensional array. Write a C program to count and display positive, negative, odd, and even numbers in an array.
3. WAP to read a list of integers and store it in a single dimensional array. Write a C program to find the frequency of a particular number in a list of integers.
4. WAP that reads two matrices A (m x n) and B (p x q) and computes the product A and B. Read matrix A and matrix B in row major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

**Experiment 6: Functions**

1. Develop a recursive and non-recursive function FACT(num) to find the factorial of a number, n!, defined by FACT(n) = 1, if n = 0. Otherwise, FACT(n) = n \* FACT(n-1). Using this function, write a C program to compute the binomial coefficient. Tabulate the results for different values of n and r with suitable messages.
2. Develop a recursive function GCD (num1, num2) that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers.
3. Develop a recursive function FIBO (num) that accepts an integer argument. Write a C program that invokes this function to generate the Fibonacci sequence up to num.
4. Develop a C function ISPRIME (num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.
5. Develop a function REVERSE (str) that accepts a string argument. Write a C program that invokes this function to find the reverse of a given string.

**Experiment 7: Structures and Union**

1. Write a C program that uses functions to perform the following operations:
   1. Reading a complex number.
   2. Writing a complex number.
   3. Addition and subtraction of two complex numbers

Note: represent complex number using a structure.

1. Write a C program to compute the monthly pay of 100 employees using each employee‗s name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.
2. Create a Book structure containing book\_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.
3. Create a union containing 6 strings: name, home\_address, hostel\_address, city, state and zip. Write a C program to display your present address.

**Experiment 8: Pointers**

1. Declare different types of pointers (int, float, char) and initialize them with the addresses of variables. Print the values of both the pointers and the variables they point to.
2. Perform pointer arithmetic (increment and decrement) on pointers of different data types. Observe how the memory addresses change and the effects on data access.
3. Write a function that accepts pointers as parameters. Pass variables by reference using pointers and modify their values within the function.

**Experiment 9: File Handling in C**

1. Write a program to create a new file and write text into it.
2. Open an existing file and read its content character by character, and then close the file.
3. Open a file, read its content line by line, and display each line on the console.

**Experiment 10: Dynamic Memory Allocation**

1. Write a program to create a simple linked list in C using pointer and structure.
2. Write a program to insert item in middle of the linked list.

**Experiment 11: Bitwise Operator**

1. Write a program to apply bitwise OR, AND and NOT operators on bit level.
2. Write a program to apply left shift and right shift operator.

**Experiment 12: Preprocessor and Directives in C**

1. Write a program to define some constant variable in preprocessor.
2. Write a program to define a function in directives.

**Experiment 13: Macros in C**

1. Write a program to define multiple macro to perform arithmetic functions.

**Experiment 14: Static Library in C**

1. Write a program to create a static library for performing arithmetic functions.
2. Write a program to use static library in other program.

**Experiment 15: Shared Library in C**

1. Write a program to create a shared library for performing arithmetic functions.
2. Write a program to use shared library in other program.

**Total Lab hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. B. W. Kernighan, and D. M. Ritchie, "The C programming language", 2nd Edition, Prentice Hall, 1988. 2. P. J. Deitel, and H. M. Deitel, "C: How to program", 8th Edition, Pearson Education, 2015. |
| **Reference books** | 1. B. S. S. Gottfried, "Schaum's Outline of Programming with C", 2nd Edition, McGraw-Hill, 1996. 2. P. V. D. Linden, "Expert C Programming-Deep C Secrets", Pearson Education, 1994. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG1126** | **Linux Lab** | | 0 | 0 | 2 | 2 |
| **Total Units to be Covered: 12** | | **Total Contact Hours: 60** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Identify different Linux distributions available.
2. Explain the functionality of basic file operations and file viewing/editing commands.
3. Utilize basic commands to navigate the file system, create and manage files/directories, and view/edit file content.
4. Analyse complex file and directory operations, such as searching for files based on specific criteria or patterns.
5. Create shell scripts that automate tasks such as printing messages, performing arithmetic operations, and manipulating strings.

**Course Outcomes**

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

1. Understand the functionality and purpose of different file operations, user management commands, and system information commands in Linux.
2. Apply the appropriate commands to navigate the file system, manage files/directories, view/edit files, manage users, and gather system information.
3. Analyse and identify potential issues or improvements in shell scripts by examining their logic, structure, and performance.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | **-** |
| **Average** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**List of Experiments**

**Experiment 1: Install virtual player and Linux.**

**Theory:**

History of UNIX, The UNIX philosophy, GUI, Overview of the Linux Operating System, Unix commands, Introduction to VI editor

**Lab:**

Step 1: Install VMware Workstation 15 Player or VirtualBox by Oracle.

1. VMware Workstation 15 Player is a virtualization software that allows you to run multiple operating systems on a single physical machine. You can download it from the VMware website and follow the installation instructions.
2. VirtualBox is another popular virtualization software that provides similar functionality. You can download it from the Oracle website and install it on your computer.

Step 2: Download an installation .iso for a Linux distribution like Ubuntu, Fedora, or any other of your choice.

1. An .iso file is a disk image of the Linux distribution you want to install. It contains all the necessary files for the operating system.
2. You can download the .iso file from the official website of the Linux distribution you prefer. For example, if you want to use Ubuntu, you can visit the Ubuntu website and download the .iso file for the desired version.

Step 3: Install the .iso from the virtual VBox or VMware Workstation.

1. Launch VMware Workstation or VirtualBox on your computer.
2. Create a new virtual machine by clicking on the "New" or "Create" button, depending on the software you are using.
3. Follow the on-screen instructions to set up the virtual machine. This includes specifying the name, location, and specifications for the virtual machine, such as the amount of RAM and storage allocated.
4. During the setup process, you will be prompted to select the installation .iso file. Browse and select the .iso file you downloaded in Step 2.
5. Complete the virtual machine setup by following the remaining on-screen instructions. This may involve selecting the operating system type, configuring network settings, and specifying storage options.
6. Once the setup is complete, start the virtual machine. It will boot from the .iso file and begin the Linux installation process. vii. Follow the installation prompts and provide any required information, such as username, password, and partitioning options.

viii. Once the installation is complete, you will have a fully functional Linux virtual machine running within VMware Workstation or VirtualBox.

**Experiment 2: Practice some basic commands on Linux**

**Theory:** Overview of file systems in Linux, File permissions and ownership, modifying file permissions and ownership, Creating and editing files using text editors

**Lab**:

1. Basic Navigation Commands: Teach essential commands for navigating the file system, such as:

ls (list): Display the contents of a directory. cd (change directory): Move between directories.

Pwd (print working directory): Show the current directory. mkdir (make directory): Create new directories.

rmdir (remove directory): Remove empty directories.

1. File Operations: Introduce fundamental file operations using the following commands: touch: Create new files. cp (copy): Copy files and directories. mv (move): Move or rename files and directories.

rm (remove): Delete files and directories.

1. File Viewing and Editing: Introduce commands for viewing and editing files, such as: cat: Display file contents. less or more: View files with pagination. head and tail: Show the beginning or end of a file.

nano or vim: Basic text editors.

1. User Management: Discuss commands for user management tasks, including: whoami: Display the current user. who: Show users currently logged in. passwd: Change the password for the current user.

sudo (superuser do): Execute commands with administrative privileges.

1. System Information: Introduce commands for gathering system information, such as: uname: Display system information. df (disk free): Show disk space usage. top or htop: Monitor system processes. history: View command history.

**Experiment 3: Files and Directories commands**

**Theory:** File Manipulation Commands, File Compression and Archiving, File Searching, File System Navigation and Management, File Transfer

**Lab:**

* 1. Working with Files: touch: Create an empty file or update the access/modify timestamps of an existing file. cp: Copy files and directories.

mv: Move or rename files and directories. rm: Remove files and directories. cat: Concatenate and display the contents of a file. less: Display the contents of a file one page at a time. head: Display the first few lines of a file. tail: Display the last few lines of a file.

* 1. File Permissions and Ownership:

Explain the ls -l command to display detailed file information, including permissions and ownership.

Discuss the three sets of permissions: owner, group, and others.

Explain the chmod command to modify file permissions.

Discuss the chown and chgrp commands to change file ownership and group.

* 1. Advanced File and Directory Operations: find: Search for files and directories based on various criteria. grep: Search for specific patterns within files.

tar: Archive files and directories into a single file. gzip/gunzip: Compress and decompress files.

ln: Create hard and symbolic links.

**Experiment 4: Shell Programming**

**Theory:** Introduction to BASH shell scripting, Basics of Shell Scripting, Types of Shell, Shell variable, Shell Keywords, Basic Operator, Positional Parameters

**Lab:**

1. Write a simple shell script that prints "Hello, World!" when executed.
2. Create a script that prompts the user to enter their name and then displays a personalized greeting.
3. Write a script that takes two numbers as input and performs various arithmetic operations like addition, subtraction, multiplication, and division.
4. Create a script that asks the user to enter their age and displays a message based on whether they are eligible to vote or not.

**Experiment 5: Shell Programming**

**Theory:** Command Line Argument, Array, Conditional Statements, Decision Making,

**Lab:**

1. Write a script that takes a number as input and checks whether it is a prime number or not.
2. Write a script that calculates the sum of the digits of a given number.
3. Create a script that checks whether a given number is an Armstrong number or not.

**Experiment 6: Shell Programming**

**Theory:** Shell Loops, Loop control, IO Redirections, Shell Function, Regular Expressions, Script Debugging and Troubleshooting

**Lab:**

1. Write a script that checks whether a given number is a palindrome or not. A palindrome number reads the same backward as forward.
2. Write a script that calculates the greatest common divisor (GCD) and the least common multiple (LCM) of two given numbers.

iii. Create a script that takes multiple numbers as input and sorts them in ascending or descending order.

**Experiment 7: Shell Programming**

**Theory:** Introduction to Processes, Process states, and process hierarchy, Process Management Commands: Viewing and monitoring running processes, Terminating or killing processes, Process Prioritization and Scheduling.

**Lab:**

1. Write a script that takes a filename as input and checks if it exists. If the file exists, display its content; otherwise, prompt the user to create the file.
2. Create a script that prints the numbers from 1 to 10 using a loop.
3. Write a script that takes a filename as a command line argument and counts the number of lines, words, and characters in that file.
4. Create a script that defines a function to calculate the factorial of a given number and call that function with different inputs.

**Experiment 8: Shell Programming**

**Theory:** Process Control and Signals, Process Monitoring and Resource Usage, Process

Communication, Process Synchronization, Background Processes and Job Control, System Monitoring and Logging

**Lab:**

1. Write a script that checks the file permissions of a given file and displays whether it is readable, writable, or executable by the current user.
2. Create a script that prompts the user to enter a string and then performs operations like string length, string concatenation, and string comparison.
3. Write a script that searches for a specific pattern in a given file and displays the matching lines.
4. Create a script that displays various system information like the current date and time, logged-in users, system uptime, etc.

**Experiment 9: Shell Programming**

**Theory:** System Performance Monitoring, System Security and User Management.

**Lab:**

1. Write a script that renames all files in a directory by adding a prefix or suffix to the filenames.
2. Create a script that searches for files in a specified directory and its subdirectories, based on certain criteria like file extension or file size.
3. Write a script that generates the Fibonacci series up to a given number, using loops or recursive functions.

**Experiment 10: Shell Programming**

**Theory:** Writing modular and reusable code, Script optimization

**Lab:**

1. Write a script that takes a string as input and calculates its length. ii. Create a script that takes a string as input and prints its reverse. iii. Write a script that prompts the user to enter two strings and concatenate them together.

**Experiment 11: Shell Programming**

**Theory:** Interacting with Users: Interactive shell scripts, Parsing and Processing Data Formats, Interacting with Databases

**Lab:**

1. Write a script that takes a sentence as input and splits it into individual words.
2. Create a script that checks whether a given string is a palindrome or not.

**Experiment 12: Building a Rule-Based Expert System using Shell Scripting**

**Theory:**

Process Automation and Job Scheduling: Automating repetitive tasks using shell scripts, System Administration Scripts, Managing services and daemons.

**Lab:**

Objective: The objective of this lab exercise is to build a simple rule-based expert system using shell scripting. The expert system will provide recommendations based on a set of predefined rules.

Instructions:

1. Create a shell script named "expert\_system.sh".
2. Implement a set of rules using conditional statements (if-elif-else) within the script. Each rule should check for specific conditions and provide a corresponding recommendation.
3. Example rules:
   1. If the user is experiencing fever, recommend taking a fever reducer medication.
   2. If the user has a sore throat, recommend gargling with warm saltwater.
   3. If the user has a cough and congestion, recommend drinking warm fluids and taking cough syrup.
   4. Feel free to add more rules based on your desired expert system topic.

1. Prompt the user to input their symptoms.
2. Based on the user's input, evaluate the rules one by one and display the appropriate recommendation(s) for the symptoms identified.
3. If none of the rules match the user's symptoms, provide a general recommendation or message.
4. Test the expert system by running the script and providing different sets of symptoms to observe the recommendations.
5. Modify the rules or add new rules as needed to refine the expert system's behavior.
6. Document the logic and rules implemented in the script, along with any modifications or additions made.
7. Write a summary report discussing the challenges faced, observations made, and improvements that can be made to enhance the expert system's functionality.

**Total Lab hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Dayanand Ambawade, and Deven Shah, "Linux Labs and Open Source Technologies", Dreamtech Press, 2014. 2. Paul W Browning, "101 Labs - Linux LPIC1: Includes Linux Essentials", Reality Press Ltd, 2019. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:**

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG1027** | **Problem Solving** | | 2 | 0 | 0 | 2 |
| **Total Units to be Covered: 4** | | **Total Contact Hours: 30** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Learn problem investigation strategies.
2. Understand the tactics for solving different problems.
3. Apply the learnt tactics to typical problems.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Identify various problem-solving strategies.

**CO2:** Infer the potential tactics for problem solving.

**CO3:** Develop appropriate tactics for solving the posed problems.

**CO4:** Analyse the solutions towards efficacy.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO4** | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **Average** | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: 7 Lecture Hours**

Thinking on a problem:Problems vs Exercises, the levels of problem solving, problem classification. Theoretical frameworks for problem solving:Polya’s theory, cognitive problem-solving, functional fixedness and mental set, means-end analysis, divergent and convergent thinking, backdrop of theoretical frameworks, and relevant examples

**Unit II: 8 Lecture Hours**

Investigating problems:Psychological Strategies, strategies for Getting Started, Methods of Argument, other strategies; drawing pictures.

Heuristics for problem solvingheuristics and heuristic reasoning, Symmetry – geometric and algebraic symmetry, the extreme principle, the Pigeonhole principle – basic, intermediate, and advanced, sample problems.

Further tactics for problem solving:Invariants, Coloring proofs, the Box principle, sample problems.

**Unit III: 7 Lecture Hours**

Crossover tactics: Graph Theory - Connectivity and Cycles, Eulerian and Hamiltonian Paths, Complex Numbers - Basic Operations, roots of unity, Generating Functions - Recurrence Relations, partitions, problems based on crossover tactics.

Algebra: Polynomials – Polynomial operations, Inequalities – Fundamental ideas, the AM-GM inequality, Massage, Cauchy-Schwarz, and Chebyshev inequalities, problems based on algebraic tactics.

**Unit IV: 8 Lecture Hours**

Combinatorics: Partitions and Bijections – counting subsets, The Principle of Inclusion-Exclusion – Count the Complement, PIE with Sets, PIE with Indicator Functions, sample problems based on combinatorics.

Number Theory: Primes and divisibility – The Fundamental Theorem of Arithmetic, GCD, LCM, and the Division Algorithm, Congruence – Fermat's Little Theorem, Number Theoretic Functions – divisor sums, Phi and Mu, Diophantine Equations, sample problems based on number theory.

**Puzzles and challenging problems:** Algorithmic puzzles and games – weighing problems, e.g., ranking weights, red, white, and blue weights, fair division problems, The tower of Hanoi, Crossing a river,the puzzle of twins, the puzzle of thinking about oneself, the bridge problem, fake coin detection, Handshake problem.

**Total lecture Hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. G. Polya, "How to solve it: A new aspect of mathematical method", 2nd Edition, Princeton university press, 2004. 2. P. Zeitz, "The art and craft of problem solving", 3rd Edition, John Wiley & Sons, 2017. 3. R. Backhouse, "Algorithmic problem solving", John Wiley & Sons, 2011. |
| **Reference books** | 1. A. Engel, "Problem-solving strategies", Springer, 2008. 2. M. Gardner, and D. Richards, "The colossal book of short puzzles and problems", W. W. Norton & Company, 2005. 3. D. D. Riley, and K. A. Hunt, "Computational thinking for the modern problem solver", Chapman and Hall/CRC, 2014 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Components | IA | MID SEM | End Sem | Total |
| Weightage (%) | 50% | 20% | 30% | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SSEN0101** | **Environment Sustainability & Climate Change** | | 2 | 0 | 0 | 2 |
| **Total Units to be Covered: 9** | | **Total Contact Hours: 30** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To develop a critical understanding of the nature, cause and impact of human activities on the environment.
2. Critically engage with concepts of ecosystems, biodiversity and sustainability.
3. Research, analyse, identify problems, develop insights, and frame sustainable solutions to living issues faced by the global and local communities.
4. Learning by doing, engaging, exploring and experimenting.

**Course Outcomes**

1. Understand the concepts of ecology, sustainability, climate change and environment related to everyday life.
2. Distinguish and relate different types of biodiversity and natural resources and their impact on sustainable development.
3. Analyse various aspects of environment and adopt eco-friendly technologies to facilitate conservation and regeneration of natural resource.
4. Buildenvironmental awareness through a wide range of curricular and co-curricular activities at the University and later in a professional/vocational practice.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **CO4** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **Average** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium) 3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Humans and the Environment 2 lecture hours**

The man-environment interaction: Humans as hunter-gatherers; Mastery of fire; Origin of agriculture; Emergence of city-states; Great ancient civilizations and the environment; Middle Ages and Renaissance; Industrial revolution and its impact on the environment; Population growth and natural resource exploitation; Global environmental change.

The emergence of environmentalism: Anthropocentric and eco-centric perspectives (Major thinkers); The Club of Rome- Limits to Growth; UN Conference on Human Environment 1972; World Commission on Environment and Development and the concept of sustainable development; Rio Summit and subsequent international efforts.

**Unit II: Natural Resources and Sustainable Development 3 lecture hours**

Overview of natural resources: Definition of resource; Classification of natural resources- biotic and abiotic, renewable and non-renewable.

Biotic resources: Major type of biotic resources- forests, grasslands, wetlands, wildlife and aquatic (fresh water and marine); Microbes as a resource; Status and challenges.

Water resources: Types of water resources- fresh water and marine resources; Availability and use of water resources; Environmental impact of over-exploitation, issues and challenges; Water scarcity and stress; Conflicts over water.

Soil and mineral resources: Important minerals; Mineral exploitation; Environmental problems due to extraction of minerals and use; Soil as a resource and its degradation.

Energy resources: Sources of energy and their classification, renewable and non-renewable sources of energy; Conventional energy sources- coal, oil, natural gas, nuclear energy; Non-conventional energy sources- solar, wind, tidal, hydro, wave, ocean thermal, geothermal, biomass, hydrogen and fuel cells; Implications of energy use on the environment.

Introduction to sustainable development: Sustainable Development Goals (SDGs)- targets and indicators, challenges and strategies for SDGs.

**Unit III: Environmental Issues: Local, Regional and Global 3 lecture hours**

Environmental issues and scales: Concepts of micro-, meso-, synoptic and planetary scales; Temporal and spatial extents of local, regional, and global phenomena.

Pollution: Impact of sectoral processes on Environment, Types of Pollution- air, noise, water, soil, municipal solid waste, hazardous waste; Transboundary air pollution; Acid rain; Smog.

Land use and Land cover change: land degradation, deforestation, desertification, urbanization.

Biodiversity loss: past and current trends, impact.

Global change: Ozone layer depletion; Climate change.

**Unit IV: Conservation of Biodiversity and Ecosystem 4 lecture hours**

Biodiversity and its distribution: Biodiversity as a natural resource; Levels and types of biodiversity; Biodiversity in India and the world; Biodiversity hotspots; Species and ecosystem threat categories.

Ecosystems and ecosystem services: Major ecosystem types in India and their basic characteristics- forests, wetlands, grasslands, agriculture, coastal and marine; Ecosystem services- classification and their significance.

Threats to biodiversity and ecosystems: Land use and land cover change; Commercial exploitation of species; Invasive species; Fire, disasters and climate change.

Major conservation policies: in-situ and ex-situ conservation approaches; Major protected areas; National and International Instruments for biodiversity conservation; the role of traditional knowledge, community-based conservation; Gender and conservation.

**Unit V: Environment Pollution and Health 4 lecture Hours**

Understanding pollution: Production processes and generation of wastes; Assimilative capacity of the environment; Definition of pollution; Point sources and non-point sources of pollution.

Air pollution: Sources of air pollution; Primary and secondary pollutants; Criteria pollutants- carbon monoxide, lead, nitrogen oxides, ground-level ozone, particulate matter and sulphur dioxide; Other important air pollutants- Volatile Organic compounds (VOCs), Peroxyacetyl Nitrate (PAN), Polycyclic aromatic hydrocarbons (PAHs) and Persistent organic pollutants (POPs); Indoor air pollution; Adverse health impacts of air pollutants; National Ambient Air Quality Standards.

Water pollution: Sources of water pollution; River, lake and marine pollution, groundwater pollution; water quality Water quality parameters and standards; adverse health impacts of water pollution on human and aquatic life.

Soil pollution and solid waste: Soil pollutants and their sources; Solid and hazardous waste; Impact on human health.

Noise pollution: Definition of noise; Unit of measurement of noise pollution; Sources of noise pollution; Noise standards; adverse impacts of noise on human health.

Thermal and Radioactive pollution: Sources and impact on human health and ecosystems.

**Unit VI: Climate Change Impact Adaptation and Mitigation 4 lecture Hours**

climate change from greenhouse gas emissions– past, present and future; Projections of global climate change with special reference to temperature, rainfall, climate variability and extreme events; Importance of 1.5 °C and 2.0 °C limits to global warming; Climate change projections for the Indian sub-continent.

Impacts, vulnerability and adaptation to climate change: Observed impacts of climate change on ocean and land systems; Sea level rise, changes in marine and coastal ecosystems; Impacts on forests and natural ecosystems; Impacts on animal species, agriculture, health, urban infrastructure; the concept of vulnerability and its assessment; Adaptation vs. resilience; Climate-resilient development; Indigenous knowledge for adaptation to climate change.

Mitigation of climate change: Synergies between adaptation and mitigation measures; Green House Gas (GHG) reduction vs. sink enhancement; Concept of carbon intensity, energy intensity and carbon neutrality; National and international policy instruments for mitigation, decarbonizing pathways and net zero targets for the future; Energy efficiency measures; Renewable energy sources; Carbon capture and storage, National climate action plan and *Intended Nationally Determined Contributions* (INDCs); Climate justice.

**Unit VII: Environment Management 4 Lecture Hours**

Introduction to environmental laws and regulation: Constitutional provisions- Article 48A, Article 51A (g) and other derived environmental rights; Introduction to environmental legislations on the forest, wildlife and pollution control.

Environmental management system: ISO 14001

Life cycle analysis; Cost-benefit analysis

Environmental audit and impact assessment; Environmental risk assessment

Pollution control and management; Waste Management- Concept of 3R (Reduce, Recycle and Reuse) and sustainability; Ecolabeling /Ecomark scheme

**Unit VIII: Environment Treaties and Legislation**   **4 Lecture Hours**

1. An overview of instruments of international cooperation; bilateral and multilateral agreements; conventions and protocols; adoption, signature, ratification and entry into force; binding and non-binding measures; Conference of the Parties (COP)

2. Major International Environmental Agreements: Convention on Biological Diversity (CBD); Cartagena Protocol on Biosafety; Nagoya Protocol on Access and Benefit-sharing; Convention on International Trade in Endangered Species of 6ild Flora and Fauna (CITES); Ramsar Convention on Wetlands of International Importance; United Nations Convention to Combat Desertification (UNCCD); Vienna Convention for the Protection of the Ozone Layer; Montreal Protocol on Substances that Deplete the Ozone Layer and the Kigali Amendment; Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and their Disposal; Rotterdam Convention on the Prior Informed Consent Procedure for Certain Hazardous Chemicals and Pesticides in International Trade; Stockholm Convention on Persistent Organic Pollutants; Minamata Convention on Mercury; United Nations Framework Convention on Climate Change (UNFCCC); Kyoto Protocol; Paris Agreement; India’s status as a party to major conventions

3. Major Indian Environmental Legislations: The Wild Life (Protection) Act, 1972; The Water (Prevention and Control of Pollution) Act, 1974; The Forest (Conservation) Act, 1980; The Air (Prevention and Control of Pollution) Act, 1981; The Environment (Protection) Act, 1986; The Biological Diversity Act, 2002; The Scheduled Tribes and Other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006; Noise Pollution (Regulation and Control) Rules, 2000; Industry-specific environmental standards; Waste management rules; Ramsar sites; Biosphere reserves; Protected Areas; Ecologically Sensitive Areas; Coastal Regulation Zone; Status phase-out of production and consumption of Ozone Depleting Substances by India; National Green Tribunal; Some landmark Supreme Court judgements

Major International organisations and initiatives: United Nations Environment Programme (UNEP), International Union for Conservation of Nature (IUCN),World Commission on Environment and Development (WCED), United Nations Educational, Scientific and Cultural Organization (UNESCO), Intergovernmental Panel on Climate Change (IPCC), and Man and the Biosphere (MAB) programme.

**Unit IX: Living Lab Case Studies and Field Work 2 Lecture Hours**

The students are expected to be engaged in some of the following or similar identified activities:

1. Discussion on one national and one international case study related to the environment and sustainable development.
2. Field visits to identify local/regional environmental issues, make observations including data collection and prepare a brief report.
3. Documentation of campus biodiversity.
4. Campus environmental management activities such as solid waste disposal, water management, and sewage treatment

**Total lecture Hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Michael Herbert Fisher, An environmental history of India: from earliest times to the twenty-first century. Cambridge, United Kingdom; New York, Ny: Cambridge University Press, 2018. 2. D. R. Headrick, Humans versus nature: a global environmental history. New York, Ny: Oxford University Press, 2020. 3. William P.Cunningham and Mary A. Cunningham Environmental Science: A Global Concern, Publisher (Mc-Graw Hill, USA), 2015 4. Gilbert M. Masters and W. P, An Introduction to Environmental Engineering and Science, Ela Publisher (Pearson), 2008 5. R. Rajagopalan, Environmental Studies: From Crisis to Cure. India: Oxford University Press.University Grants Commission 11 **(**2011). |
| **Reference books** | 1. William P. Cunningham and Mary A. Cunningham Environmental Science: A global concern, Publisher (Mc-Graw Hill, USA), 2015. 2. Shonil Bhagwat, Conservation and Development in India: Reimagining Wilderness, Earthscan Conservation and Development, Routledge , (Editor) (2018). 3. G. M. Masters, & W. P Ela, *Introduction to environmental engineering and science* (No. 60457). Englewood Cliffs, NJ: Prentice Hall (2008). 4. G. T. Miller, & S. Spoolman, Environmental Science. Cengage Learning (2015). 5. Central Pollution Control Board Web page for various pollution standards. <https://cpcb.nic.in/standards/> 6. V. K. Ahluwalia, *Environmental Pollution, and Health*. The Energy and Resources Institute (TERI), 2015.**University Grants Commission 13** 7. A. Denle, H. Azadi, J. Arbiol Global assessment of technological innovation for climate change adaptation and mitigation in developing world, Journal of Environmental Management, 2015, 161 (15): 261-275. 8. Richard A. Marcantonio, Marc Lame, Environmental Management: Concepts and Practical Skills. Cambridge University Press,2022, **University Grants Commission 15** 9. UNEP (2007) Multilateral Environmental Agreement Negotiator’s Handbook, University of Joensuu, ISBN 978-952-458-992-5 10. Ministry of Environment, Forest and Climate Change (2019) A Handbook on International Environment Conventions & Programmes. https://moef.gov.in/wp- content/uploads/2020/02/convention-V-16-CURVE-web.pdf |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme: Continous assessment**

|  |  |
| --- | --- |
| **Components** | **Continous Assesment** |
| Weightage (%) | 100% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **PHYS1036** | **Physics for Computer Engineers** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 4** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To demonstrate the principles of LASER and its applications in holography as well as in fiber-optic communications.

2. To determine gradient of scalar fields and divergence & curl vector fields.

3. To develop understanding of electromagnetics, which forms the basis of several contemporary communication systems such as fiber optics communication and it, is also a prerequisite for forthcoming semesters.

4. To utilize fundaments of quantum mechanics in various areas of Material Science and engineering.

5. To understand and apply semiconductor materials in various applications.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Understand the significance of lasers and its application in holography and optical fiber communication.

**CO2**. Illustrate the electric field for different charge geometries.

**CO3.** Outline the magnetic field due to different current geometries.

**CO4.** Utilize the fundamentals of Quantum Mechanics and analyse the behaviour of particle in a box.

**CO5.** Apply various applications of semiconductor materials in different instruments.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 2** | 3 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 3** | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 4** | 3 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 5** | 3 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **Average** | 2.8 | 1.8 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Lasers & Fibre Optics 14 Lecture Hours**

Introduction, Spontaneous and Stimulated emission of radiation, Relation b/w Einstein’s A and B coefficients, Population inversion & types of pumping, Main components of a Laser, Construction & working of Ruby Laser and its applications, Construction & working of Helium-Neon laser and its applications. Holography: Elementary idea of holography and constructive and reconstructive of holography.

Fundamental ideas about optical fiber, Types of fibers, Acceptance angle and cone, Numerical aperture, Propagation mechanism and communication in optical fiber, Attenuation and losses.

**Unit II: Electro-Magnetics 16 Lecture Hours**

**Electro-statics**: Coordinate systems, Del operator, Gradient, Divergence, Divergence Theorem, Stoke’s Theorem, Introduction to electrostatics, calculation of electric field, potential and energy due to charge distribution by vector approach, Gauss law electric flux density. Polarization in Dielectrics, Bound charges, Dielectric Constant and strength, Continuity equation and relaxation time Boundary Conditions.

**Magneto-statics:** Introduction, Biot-Savart’s law, Ampere’s Circuit Law; Applications, Magnetic flux density, Faraday’s Law, Transformer and motional EMF. Displacement current, Maxwell’s Equations in Final form.

**Unit III: Quantum Mechanics 15 Lecture Hours**

Introduction to Quantum Mechanics, photoelectric effect, Compton Effect, Pair production & Annihilation, Wave particle duality, De Broglie waves, Davisson Germer experiment, phase and group velocities and their relations, Thought experiment- Heisenberg’s Gamma ray microscope, Uncertainty principle and its applications, Wave function and its interpretation, Normalization, Schrodinger time independent & dependent wave equations, Particle in a 1-D box; generalization to 3-D box.

**Unit IV: Semiconductor Physics 15 Lecture Hours**

P and N type semiconductors, Energy Level Diagram, Conductivity and Mobility, Concept of Drift velocity, Hall effect, Barrier Formation in PN Junction Diode, Static and Dynamic Resistance, Current Flow Mechanism in Forward and Reverse Biased Diode, Avalanche breakdown, Zener breakdown, Two-terminal Devices and their Applications: Half-wave Rectifiers, Full-wave Rectifiers, Ripple Factor and Rectification Efficiency, Zener Diode and Voltage Regulation, Principle and structure of LED, Photodiode and Solar Cell

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. H. K. Malik, A. K. Singh, "Engineering Physics", 2nd Edition, McGraw Hill Education, 2017. 2. A. Beiser, "Concepts of Modern Physics", 4th Edition, McGraw Hill Education, 2018. |
| **Reference books** | 1. D. J. Griffith, "Introduction to Electromagnetics", 4th Edition, Cambridge University Press, 2020. 2. A. Ghatak, "Optics", 7th Edition, McGraw Hill Education, 2020. 3. V. Sahni, and D. Goswami, "Nano Computing", McGraw Hill Education Asia Ltd., 2008. 4. M. N. O. Sadiku, "Elements of Electromagnetics", 3rd Edition, Oxford University Press, 2020. 5. C. T. Bhunia, "Introduction to Quantum Computing", New Age International Publishers, 2010. 6. S. M. Sze, "Semiconductor Devices: Physics and Technology", 2nd Edition, John Wiley & Sons, 2001. 7. S. Salivahanan, and N. S. Kumar, "Electronic Devices & circuits", 5th Edition, McGraw Hill, 2022. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Physics Lab for Computer Engineers**

**List of Experiments**

**Experiment 1:**

To determine the frequency of AC mains by using a sonometer.

**Experiment 2:**

To study the Hall effect and hence determine the Hall coefficient (Rh) and carrier density (n) of a given semiconductor material.

**Experiment 3:**

(a) To study the induced emf as a function of velocity of the magnet passing through the coil (Faraday’s Law).

(b) To study the charge delivered due to electromagnetic induction.

**Experiment 4:**

To study the variation of magnetic field with distance along the axis of a current carrying circular coil and hence estimate the radius of the coil.

**Experiment 5:**

To plot the characteristics of photocurrent vs voltage at different frequency.

**Experiment 6:**

To determine the Numerical Aperture of an optical fibre and study about the bending losses.

**Experiment 7:**

To study the laser beam diffraction.

**Experiment 8:**

Study of both the current - voltage characteristic and the power curve to find the maximum power point (MPP) and efficiency of a solar cell.

**Experiment 9:**

To find the Planck’s constant by using LEDs.

**Experiment 10:**

To determine the energy band gap of a given semiconductor by using Four-Probe Method.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. H. Singh, and P. S. Hemne, "Practical Physics", S. Chand & Company Ltd., 2022. 2. S. L. Kakani, and S. Kakani, "Applied Physics-Theory & Practicals", Viva Books, 2014. 3. C. L. Arora, "Practical Physics", S. Chand & Company Ltd., 2010. |
| **Reference books** | 1. S. L. Gupta, and V. Kumar, "Practical Physics", 4th Edition, Vol. 1, Pragati Prakashan, 2017. 2. I. Prakash, R. Krishna, and A. K. Jha, "Practical Physics", Vol. 1, Kitab Mahal, 2011. 3. P. R. S. Kumar, "Practical Physics", Prentice Hall India Learning Pvt. Ltd., 2011. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SFLS0001** | **Managing Self** | |  |  |  | 2 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**SEMESTER II**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Name** | | **L** | **T** | **P** | **C** |
| **CSEG1043** | **Data Structures and Algorithms** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Programming in C-CSEG 1025** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Provide a clear understanding of the importance of data structures in organizing and manipulating data efficiently.
2. To introduce students to fundamental data structures and the properties, memory management, and basic operations of each data structure.
3. To offer practical experience in implementing data structures, common sorting, and searching algorithms.
4. Emphasize the use of data structures as tools for algorithmic problem-solving and apply their knowledge of data structures to solve real-world problems.

**Course Outcomes**

Upon completion of the course, the students will be able to

1. State the significance and properties of the fundamental data structures.
2. Implement common data structures while ensuring proper memory management and error handling.
3. Illustrate expertise in understanding the common sorting and searching techniques with their complexities and implement them.
4. Analyse real-world problems by understanding the trade-offs involved in identifying the appropriate data structure(s) based on problem requirements and using them to solve the problems efficiently.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 2** | 1 | 1 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | 1 | 1 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** |
| **CO 4** | 1 | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | 1 | 1 | 2 | 1.25 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.25 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: INTRODUCTION 12 Lecture Hours**

Overview, classification, and Importance of data structures in Programming/problem-solving, Basic terminology, and concepts: elements, operations, storage, memory allocation, garbage collection, and compaction. Iterative & Recursive approaches. Basic analysis of algorithms: Amortized analysis and Asymptotic Analysis. Array: Memory representation (1D and 2D), Array operations: insertion, deletion, searching. Applications of Array. Structure: Nested Structure, Function pointer as member of a structure, Self-referential structure. Anonymous Unions, ADTs.

**Unit II: LINKED LIST 10 Lecture Hours**

Singly-Linked List, Doubly-Linked List, Circular Linked List, Header List, and its operations. Sentinel node. Generalized Linked List. Skip List. Applications of Linked Lists: polynomial manipulation, implementation of other data structures.

**Unit III: STACK & QUEUE 10 Lecture Hours**

Stack data structure and operations. Queue data structure and operations. Implementation of Stack and Queue using Array and Linked List. Circular Queue. Deque and its types. Priority Queue. Applications: Stacks (Conversion of Infix to Prefix/Postfix, Expression evaluation, a note on DFS in graph), Queues (Job scheduling, a note on BFS in graph).

**Unit IV: TREE 10 Lecture Hours**

Introduction to Tree data structure and its terminologies, Binary Tree: properties, traversal algorithms (level-order, in-order, pre-order, post-order). Threaded Binary Tree. Binary Search Trees (BST): properties, operations (insertion, deletion, searching). Balanced BSTs. AVL Tree: properties, rotations, operations (insertion, deletion). Red-Black Tree. Multi-way search Tree: properties. B-Tree: properties, operations (search, insertion, and deletion). Applications of AVL Tree and B-Tree. Binary Heaps: properties, heapify operations, Heap sorting.

**Unit V: HASH TABLE & GRAPH 10 Lecture Hours**

Hashing and hash functions. Hash table data structure: structure, collisions, collision resolution techniques, maintaining load factor. Applications of hash tables: dictionaries, symbol tables.

Introduction to graph data structure and its terminologies. Graph representations: adjacency matrix, adjacency list. Graph traversal algorithms: depth-first search, breadth-first search (BFS). Connected Components. Minimum spanning tree. Shortest path.

**Unit VI: SORTING & SEARCHING 8 Lecture Hours**

Stability and In-place properties, Internal and external sorting. Simple comparison-based sorting algorithms: bubble sort, selection sort, insertion sort. Lower bound for comparison-based sorting algorithms. Recursive implementation of merge sort, quicksort, and binary search. Complexities of common sorting and searching algorithms.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. S. Lipschutz, "Data Structures with C", Schaum's Outline Series, McGraw-Hill Education (India) Pvt. Limited, 2017. 2. Y. P. Kanetkar, "Data structures through C", 4rd Edition, New Delhi: BPB, 2022. |
| **Reference books** | 1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "Data Structures and Algorithms", New Delhi: Pearson Education, 2003. 2. E. Horowitz, and S. Sahni, "Fundamentals of Data Structures in C", 2nd Edition, Hyderabad: University Press, 2008. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Data Structures and Algorithms Lab**

**List of Experiments**

**Experiment 1:** **Basic Data Structure**

To demonstrate the use of array, structure, and union along with dynamic memory allocation.

**Experiment 2: Link List Data Structure and its Applications**

To experiment with the concept of pointers, structure, and dynamic memory allocation to realize linked lists, their types, and application.

**Experiment 3:** **Stack Data Structure**

To use arrays and linked lists to implement Stack and its applications.

**Experiment 4:** **Queue Data Structure**

To demonstrate the use of arrays and linked lists to implement different variants of Queue and its applications.

**Experiment 5:** **Trees**

To demonstrate the creation of a binary tree and working with tree traversal.

**Experiment 6: Heaps**

To create a heap data structure and implement its operations, and its applications.

**Experiment 7: Hash Tables**

To implement a hash table using various collision resolution techniques, and its applications.

**Experiment 8:** **Graphs**

To demonstrate the creation of graphs and working with graph traversal algorithms.

**Experiment 9: Sorting algorithms**

To implement common sorting algorithms.

**Experiment 10:** **Searching algorithms**

To implement various search algorithms on data structures.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. S. Lipschutz, "Data Structures with C", Schaum's Outline Series, McGraw-Hill Education (India) Pvt. Limited, 2017. 2. Y. P. Kanetkar, "Data structures through C", 4rd Edition, New Delhi: BPB, 2022. |
| **Reference books** | 1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "Data Structures and Algorithms", New Delhi: Pearson Education, 2003. 2. E. Horowitz, and S. Sahni, "Fundamentals of Data Structures in C", 2nd Edition, Hyderabad: University Press, 2008. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSAI2018** | **Elements of AIML** | | 2 | 0 | 1 | 3 |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 60** | | | | |
| **Prerequisite(s):** | **Python programming - CSEG1035** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

Students will learn the basic concepts and techniques of Artificial Intelligence and Machine Learning. They will also explore their applications.

**Course Outcomes**

On completion of this course, the students will be able to

CO1. Understand the basic concepts and techniques of Artificial Intelligence.

CO2. Understand the logic of AI algorithms for solving practical problems.

CO3. Understand the basics of Machine Learning and its types.

CO4. Assess and model real-world practical problems that can be handled by AI and ML.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** |
| **CO 2** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** |
| **CO 3** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** |
| **CO 4** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **Average** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.25 | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction 6 Lecture Hours**

Artificial Intelligence and its applications, Artificial Intelligence Techniques, Level of models, criteria of success, Intelligent Agents, Nature of Agents, Learning Agents. AI Techniques, advantages, and limitations of AI, Impact and Examples of AI, Application domains of AI.

**Unit II: Logic for AI 6 Lecture Hours**

Propositional logic, predicate logic, Resolution, Resolution in proportional logic and predicate logic, Clause form, unification algorithm

**Unit III: Introduction to machine learning 6 Lecture Hours**

Introduction to Machine Learning, Usage of datasets and how to handle them for Machine Learning Feature sets, Dataset division: test, train and validation sets, cross validation, Dimensionality Reduction Techniques: PCA, LDA, ICA

**Unit IV: Types of machine learning 6 Lecture Hours**

Introduction to Machine Learning Techniques: Supervised Learning: Regression and its types, Classification, Unsupervised Learning: Clustering, Reinforcement Learning, Semi-supervised Machine Learning

**Unit V: Applications of AI and Machine learning 6 Lecture Hours**

AI for society, women and environment, Applications of Machine Learning in Banking, Security, Healthcare, Education, Insurance Industry, Retail and Supply Chain, Transportation and Logistics, Energy and Utilities

**Total lecture Hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Artificial Intelligence by Rich and Knight, The McGraw Hill, 2017. 2. Machine Learning for Dummies, By John Paul Mueller and Luca Massaron, For Dummies, 2016. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Elements of AIML Lab**

**List of Experiments**

|  |  |
| --- | --- |
| **Experiment** 1 | Exploration of WEKA tool for Regression task. |
| **Experiment** 2 | Exploration of WEKA tool for Classification task. |
| **Experiment** 3 | Exploration of WEKA tool for Clustering task. |
| **Experiment**  4 | Write a python program to import and export data using Pandas and show the details of the dataset like number of rows, columns, first five rows, size, number of missing values, sum, average, min and max values from the numerical columns. |
| **Experiment** 5 | Using Python language do the exploratory data analysis of dataset imported in the lab 4. |
| **Experiment** 6 | Implement the missing value, and outlier handling data preprocessing techniques on the dataset imported in lab 4 or any other dataset. |
| **Experiment** 7 | Implement feature scaling and one hot encoding data preprocessing techniques on the dataset imported in lab 4 or any other dataset. |
| **Experiment** 8 | Implement Dimensionality reduction using Principal Component Analysis (PCA) method. |
| **Experiment** 9 | Implement different techniques of handling imbalanced data. |
| **Experiment** 10 | Write a Python program to demonstrate various Data Visualization Techniques using Matplotlib and Seaborn libraries. |

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. “Data Science for Business" by Foster Provost and Tom Fawcett. 2. "Python for Data Analysis" by Wes McKinney. |
| **Reference books** | 1. "Data Wrangling with Python" by Kevin Markham. 2. "Storytelling with Data" by Cole Nussbaumer Knaflic. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG1021** | **Python Programming** | | **2** | **0** | **2** | **4** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Develop a strong foundation in Python programming language, including syntax, data types, control structures, and functions, enabling students to write efficient and reliable code.
2. Understand and apply object-oriented programming (OOP) principles in Python to design and build modular, reusable, and maintainable software solutions.
3. Gain proficiency in utilizing Python libraries and modules for tasks such as data manipulation, web scraping, data analysis, and visualization, empowering students to work with real-world data effectively.
4. Explore advanced topics in Python, including concurrency, file I/O, exception handling and equipping students with the skills to build robust and scalable applications.

**Course Outcomes**

**CO1.** Demonstrate proficiency in Python programming by writing code that adheres to Python syntax, utilizes appropriate data types, and implements control structures effectively.

**CO2.** Apply Python collections, such as lists, tuples, dictionaries, and sets, along with the design and implementation of reusable functions, to solve complex programming problems, demonstrating proficiency in data organization, manipulation, and modular code design.

**CO3.** Implement advanced Python features and techniques, such as modules and packages, file handling, exception handling and regular expression to create robust and reliable applications.

**CO4.** Apply object-oriented programming (OOP) concepts in Python to design and develop modular software solutions that promote code reusability and maintainability.

**CO5.** Utilize Python libraries and modules for data manipulation, analysis, and visualization, demonstrating the ability to work with real-world data sets and extract meaningful insights.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 | **-** |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | **-** |
| **Average** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1.2 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Python 5 Lecture Hours**

Introduction, Working with Python, Interactive mode and Scripting mode, Basic Syntax, Comments, Dynamic Types, Mutable and immutable data types, Python Tokens (Keyword, identifier, special symbols, literals, constants), Naming Conventions, String Values, String Methods, The format Method, String Operators, Numeric Data Types, Input & Output functions, Escape sequence characters, Operators (Arithmetic, relational, logical, assignment, bitwise, membership, identity), Operators precedence and associativity, Type function and id function, Indentation, Decision Making Statements (if, if else, elif, nested if, match statement), range function, looping structures (while loop, for loop), break, continue, pass statement, else in loops, nested loops.

**Unit II: Collections and Functions 5 Lecture Hours**

List initialization, List methods, List operations, indexing, slicing, list comprehension, Nesting in lists, tuple initialization, tuple methods, tuple operations, nesting in tuple, List vs Tuple, Set initialization, Set methods, Set operations, Dictionary initialization, Dictionary methods, nesting in Dictionary, Sorting data collections, typecasting collections, Applications of collections, Introduction to Functions, Advantages of using a function, Defining user defined function, Parameters, Function Documentation, Keyword and Optional Parameters, default argument, Variable length Arguments, Scope, Passing Collections to a Function, , Recursion, map, filter, Lambda function, Inner Functions, Passing mutable and immutable datatypes in functions.

**Unit III: File and Exception Handling 4 Lecture Hours**

File Access Modes, File handling Functions, Writing Data to a File, Reading Data from a File, Additional File Methods, With Statement, Working with Directories, Applications of File Handling, Errors vs Exceptions, The Exception Model, Exception Hierarchy, Exception Handling (try, except, else, finally), Handling Multiple Exceptions, raise, assert

Introduction to modules and packages, creating modules and packages, Standard Modules – sys, math, time, os, Regular Expressions, Meta characters.

**Unit IV: GUI Programming and data connectivity 6 Lecture Hours**

Introduction to Tkinter, Tkinter widgets – Labels, Button, Entry, Text, Canvas, scrollbars, Listbox, Combobox, Spinbox, Checkbutton, Radiobutton. Layout manager -Pack and grid. Working with menus, dialog and message boxes, Event Handling, Validating user inputs, Integrating Tkinter with Databases.

Database Connectivity: Overview of Overview of Relational and NoSQL Databases, Python Database API (DB-API 2.0), Performing CRUD operations and connecting to MongoDB, Handling transactions and error handling.

**Unit V: Class and Objects in Python 4 Lecture Hours**

OOP Concepts, Classes in Python, Creating Classes and Objects, methods in classes, Constructor, Special Methods in classes, Class Variables and Object Variables, Public and Private data members, Built-in Class Attributes, Garbage Collection, Abstract class, Inheritance, types of inheritance, Polymorphism (Function overriding, operator overloading)

**Unit VI: Data Analysis and Visualization 6 Lecture Hours**

**Numpy** – Overview, numpy Ndarray, Datatypes, Array creation, List vs Array, numpy attributes, numpy operations, Numpy Broadcasting, Numpy Functions (String, mathematical, statistical, sorting and searching), Numpy Special functions (reshape(), sum(), random(), zeros(), ones(), mean(), dot(), std(), empty(), arange(), numpy.linspace())

**Pandas** – Overview, Pandas Data Structures: Series and Data Frame, Operations on a Series (head, tail, vector operations), Data Frame operations( create, display, iteration, select column, add column, delete column), Binary operations in a Data Frame (add, sub, mul, div), Matching and broadcasting operations, Handling Missing data and filling values, Data Aggregation, Comparisons, Boolean reductions, comparing Series, Combining Data Frames, Importing/Exporting Data between CSV files and Data Frames.

**Matplotlib**- Introduction, Matplotlib Pyplot, Plotting, markers, Line, Labels, Grid, Customizing plots, Creating Different Types of Plots (Line Graph, Bar chart, Histograms, Scatter Plot, Pie Chart), Creating and working with Subplots

**Total lecture Hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Martin C. Brown, "Python: The Complete Reference", 4th Edition, McGraw Hill, 2018.  2. Paul Barry, "Head First Python", 2nd Edition, O′Reilly, 2023. Latest Edition  3. Eric Matthes, Python Crash Course: A Hands-On, Project-Based Introduction to Programming, No Starch, 3rd Edition , 2023 |
| **Reference books** | 1. Luciano Ramalho, "Fluent Python", 2nd Edition, Learning Python Series, O′Reilly, 2022. Updated edition 2. R. Nageswara Rao, Core Python Programming, Dreamtech 2021. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Python Programming Lab**

**List of Experiments**

**Experiment 1: Python Installation and basic python statements**

1. Install Python and write the steps of installation and understand difference between scripting and interactive modes in IDLE.
2. Create a variable to store your age and print its type using type().
3. Declare a string variable called x and assign it the value “Hello”.
   1. Print out the value of x
4. Take different data types and print values using print function.
5. Declare these variables (x, y and z) as integers. Assign a value of 9 to x, Assign a value of 7 to y, perform addition, multiplication, division and subtraction on these two variables and Print out the result.
6. Write a program to compute the length of the hypotenuse (c) of a right triangle using Pythagoras theorem.
7. Write a program to find simple interest.
8. Write a program to find area of triangle when length of sides are given.
9. Write a program to convert given seconds into hours, minutes and remaining seconds.
10. Write a program to swap two numbers without taking additional variable.
11. Write a program to find sum of first n natural numbers.
12. Write a program to print truth table for bitwise operators (&, | and ^ operators)
13. Write a program to find left shift and right shift values of a given number.
14. Using membership operator find whether a given number is in sequence (10,20,56,78,89)

**Experiment 2: Conditional Statements**

1. Check whether the given number is divisible by 3 and 5 both.
2. Check whether a given number is multiple of five or not.
3. Find the greatest among the two numbers. If numbers are equal than print “numbers are equal”.
4. Find the greatest among three numbers assuming no two values are same.
5. Check whether the quadratic equation has real roots or imaginary roots. Display the roots.
6. Find whether a given year is a leap year or not.
7. Write a program which takes any date as input and display next date of the calendar

e.g.

I/P: day=20 month=9 year=2005

O/P: day=21 month=9 year 2005

1. Print the grade sheet of a student for the given range of cgpa. Scan marks of five subjects and calculate the percentage.

CGPA=percentage/10

CGPA range:

0 to 3.4 -> F

3.5 to 5.0->C+

5.1 to 6->B

6.1 to 7-> B+

7.1 to 8-> A

8.1 to 9->A+

9.1 to 10-> O (Outstanding)

Sample Gradesheet

Name: Rohit Sharma

Roll Number: R17234512 SAPID: 50005673

Sem: 1 Course: B.Tech. CSE AI&ML

Subject name: Marks

PDS: 70

Python: 80

Chemistry: 90

English: 60

Physics: 50

Percentage: 70%

CGPA:7.0

Grade:

**Experiment 3: Loops**

1. Find a factorial of given number.
2. Find whether the given number is Armstrong number.
3. Print Fibonacci series up to given term.
4. Write a program to find if given number is prime number or not.
5. Check whether given number is palindrome or not.
6. Write a program to print sum of digits.
7. Count and print all numbers divisible by 5 or 7 between 1 to 100.
8. Convert all lower cases to upper case in a string.
9. Print the table for a given number:

5 \* 1 = 5

5 \* 2 = 10………..

1. Write a program to print the following pattern

123454321

1234 \*4321

123 \* \* 321

12 \* \* \* 21

1 \* \* \* \* 1

1. Write a program to print the sum of the following series

1+ ½ + 1/3 + ¼ +….+1/n

**Experiment 4: String and Sets**

1. Write a program to count and display the number of capital letters in a given string.
2. Count total number of vowels in a given string.
3. Input a sentence and print words in separate lines.
4. WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

Sample Input

ABCDCDC

CDC

Sample Output

2

1. Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet (case insensitive) and display the same.

Sample Input

ABaBCbGc

Sample Output

2A

3B

2C

1G

1. Program to count number of unique words in a given sentence using sets.
2. Create 2 sets s1 and s2 of n fruits each by taking input from user and find:
3. Fruits which are in both sets s1 and s2
4. Fruits only in s1 but not in s2
5. Count of all fruits from s1 and s2
6. Take two sets and apply various set operations on them :

S1 = {Red ,yellow, orange , blue }

S2 = {violet, blue , purple}

**Experiment 5: Lists, tuples, dictionary**

1. Scan n values in range 0-3 and print the number of times each value has occurred.
2. Create a tuple to store n numeric values and find average of all values.
3. WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output.

Sample Input

N = 5

Scores= 2 3 6 6 5

Sample output

5

Note: Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

1. Create a dictionary of n persons where key is name and value is city.

a) Display all names

b) Display all city names

c) Display student name and city of all students.

d) Count number of students in each city.

1. Store details of n movies in a dictionary by taking input from the user. Each movie must store details like name, year, director name, production cost, collection made (earning) & perform the following :-
2. print all movie details
3. display name of movies released before 2015
4. print movies that made a profit.
5. print movies directed by a particular director.
6. Create a contact book where users can store, search, update, and delete contacts. Use dictionary for storing contacts.
7. Create a Todo list Manager where users can add, view, and remove tasks. Use List for storing tasks.

**Experiment 6: Functions**

1. Write a Python function to find the maximum and minimum numbers from a sequence of numbers. (Note: Do not use built-in functions.)
2. Write a Python function that takes a positive integer and returns the sum of the cube of all the positive integers smaller than the specified number.
3. Write a Python function to print 1 to n using recursion. (Note: Do not use loop)
4. Write a recursive function to print Fibonacci series upto n terms.
5. Write a lambda function to find volume of cone.
6. Write a lambda function which gives tuple of max and min from a list.

Sample input: [10, 6, 8, 90, 12, 56]

Sample output: (90,6)

1. Write functions to explain mentioned concepts:
   1. Keyword argument
   2. Default argument
   3. Variable length argument
2. Write a program to check whether all the values in a dictionary are same or not using lambda function.
3. Write a program to create two lists and generate a dictionary with keys from list1 and values from list2.

**Experiment 7: File Handling and Exception Handling**

1. Add few names, one name in each row, in “name.txt file”.
   1. Count no of names
   2. Count all names starting with vowel
   3. Find longest name
2. Store integers in a file.
   1. Find the max number
   2. Find average of all numbers
   3. Count number of numbers greater than 100
3. Assume a file city.txt with details of 5 cities in given format (cityname population(in lakhs) area(in sq KM) ):

Example:

Dehradun 5.78 308.20

Delhi 190 1484

……………

Open file city.txt and read to:

* 1. Display details of all cities
  2. Display city names with population more than 10Lakhs
  3. Display sum of areas of all cities

1. Input two values from user where the first line contains N, the number of test cases. The next N lines contain the space separated values of a and b. Perform integer division and print a/b. Handle exception in case of ZeroDivisionError or ValueError.

Sample input

1 0

2 $

3 1

Sample Output :

Error Code: integer division or modulo by zero

Error Code: invalid literal for int() with base 10: '$' 3

1. Create multiple suitable exceptions for a file handling program.
2. Write a program to create a counter to show that how many times the program is executed.

**Experiment 8: GUI and Backend Connectivity**

1. Create a simple Tkinter window with a title and fixed size.
2. Design a GUI based basic calculator for performing basic arithmetic operations.
3. Design a GUI for student registration for a course and store these details in a database. Use Tkinter for UI, SQLite/MySQL for database storage.
4. Create a GUI based task manager where users can add, edit and remove tasks. Use Tkinter (buttons, listbox), SQLite/MySQL (task storage).
5. Design a login and signup authentication system.

**Experiment 9: Classes and objects**

1.Create a class of student (name, sap id, marks[phy,chem,maths] ). Create 3 objects by taking inputs from the user and display details of all students.

2. Add constructor in the above class to initialize student details of n students and implement following methods:

1. Display() student details
2. Find Marks\_percentage() of each student
3. Display result() [Note: if marks in each subject >40% than Pass else Fail]

Write a Function to find average of the class.

3. Create programs to implement different types of inheritances.

4. Create a class to implement method Overriding.

5. Create a class for operator overloading which adds two Point Objects where Point has x & y values

e.g. if

P1(x=10,y=20)

P2(x=12,y=15)

P3=P1+P2 => P3(x=22,y=35)

**Experiment 10: Data Analysis and Visualization**

1. Create numpy array to find sum of all elements in an array.
2. Create numpy array of (3,3) dimension. Now find sum of all rows & columns individually. Also find 2nd maximum element in the array.
3. Perform Matrix multiplication of any 2 n\*n matrices.
4. Write a Pandas program to get the powers of an array values element-wise.

Note: First array elements raised to powers from second array

Sample data: {'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]}

Expected Output:

X Y Z

0 78 84 86

1 85 94 97

2 96 89 96

3 80 83 72

4 86 86 83

5. Write a Pandas program to get the first 3 rows of a given DataFrame.

*Sample Python dictionary data and list labels:*

exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],

'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],

'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

*Expected Output:*

First three rows of the data frame:

attempts name qualify score

a 1 Anastasia yes 12.5

b 3 Dima no 9.0

c 2 Katherine yes 16.5

6. Write a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.

7. Create a program to demonstrate different visual forms using Matplotlib.

**Total Lab hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Reema Thareja, "Python Programming", Oxford University Press, 2017. 2. Mark Lutz, "Learning Python", 5th ed., O’reilly publication, 2013. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **ECEG1012** | **Digital Electronics** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Demonstrate a basic understanding of digital terminology, digital components, and systems.
2. To prepare students to perform the analysis and design of various digital electronic circuits.
3. To give the students a perspective to design combinational and sequential circuits.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Understand working of logic families and logic gates.

**CO2:** Design and implement Combinational and Sequential logic circuits.

**CO3:** Understand the process of Analog to Digital conversion and Digital to Analog conversion.

**CO4:** Understand the configuration and working of Microprocessor.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | 2 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | 2 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 4** | 2 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **Average** | 2 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Fundamentals of Digital Systems and logic families 8 Lecture Hours**

Digital signals, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive-OR operations, Boolean algebra, examples of IC gates, number systems-binary, signed binary, octal hexadecimal number, binary arithmetic, one’s and two’s complements arithmetic, codes, error detecting and correcting codes, characteristics of digital lCs, digital logic families, TTL, Schottky TTL and CMOS logic, interfacing CMOS and TTL, Tri-state logic.

**Unit II: Combinational Digital Circuits 10 Lecture Hours**

Standard representation for logic functions, K-map representation, and simplification of logic functions using K-map, minimization of logical functions. Don’t care conditions, Multiplexer, De-Multiplexer/Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder, serial adder, ALU, elementary ALU design, popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, Q-M method of function realization.

**Unit III: Sequential circuits and systems 10 Lecture Hours**

A 1-bit memory, the circuit properties of Bistable latch, the clocked SR flip flop, J- K-T And D-Types flip flops, applications of flip flops, shift registers, applications of shift registers, serial to parallel converter, parallel      to     serial    converter,         ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters, counters design using flip flops, special counter IC’s, asynchronous sequential counters, applications of counters.

**Unit IV: A/D and D/A Converters 10 Lecture Hours**

Digital to analog converters: weighted resistor/converter, R-2R Ladder D/A converter, specifications for D/A converters, examples of D/A converter lCs, sample and hold circuit, analog to digital converters: quantization and encoding, parallel comparator A/D converter, successive approximation A/D converter, counting A/D converter,     dual     slope A/D converter, A/D converter using voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs.

**Unit V: Introduction to Microprocessor 7 Lecture Hours**

Introduction to Microprocessors, Architecture of 8085 and 8086, Pin Configuration and Function; Internal Register & Flag Register, Generation of Control Signals Bus Timings: Demultiplexing of Address /Data Bus; Fetch Cycle, Execute Cycle, Instruction Cycle, Instruction Timings and Operation Status, Timing Diagram. Instruction for Data Transfer. Arithmetic and Logical Operations. Branching Operation: Machine Cycle Concept; Addressing Modes.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. R. P. Jain, "Modern Digital Electronics", 4th Edition, McGraw Hill Education, 2009. 2. M. M. Mano, "Digital logic and Computer design", Pearson Education India, |
| **Reference books** | 1. A. A. Kumar, "Fundamentals of Digital Circuits", 3rd Edition, Prentice Hall India, 2016. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **MATH1065** | **Advanced Engineering Mathematics – 2** | | **3** | **1** | **0** | **4** |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 60** | | | | |
| **Prerequisite(s):** | Advanced Engineering Mathematics – 1 - MATH1059 | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The aim of the course is to prepare the students to understand and appreciate the power of mathematics as a unifying language transcending a variety of engineering and science disciplines. The focus for the designing of the syllabus has been to provide the students with insights into the mathematical concepts and their applications without much compromising mathematical rigor.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Formulate appropriate numerical and optimization schemes for the development of computational algorithms in science and engineering.

**CO2.** Understand the fundamental tools of complex analysis for addressing and solving a variety of problems viz. special functions, integral transforms, and PDEs.

**CO3.** Identify and illustrate the use of Bessel functions and Legendre polynomials in real world application.

**CO4.** Demonstrate computational implementation of integral transforms and their applications.

**CO5.** Model real-world phenomena evolving in space and time governed by linear PDEs.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 3 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | 3 | 3 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | 3 | 3 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO4** | 3 | 3 | 3 | **-** | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | 3 | 3 | 3 | **-** | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **Average** | 3 | 3 | 2.4 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit 0: Motivation** **01 Lecture Hours**

Why study this course- Relevance and Significance?

**Unit I: Numerical Methods and Optimization 13 Lecture Hours**

Bisection and Newton-Raphson methods, Gauss Elimination and Gauss-Seidel methods, Finite difference operators, Interpolation with equal and unequal intervals, Numerical differentiation, and integration, Numerical solution of ODEs: Picard’s method, Euler’s method, Runge-Kutta fourth order method. Introduction to optimization, The Simplex method, Duality, Lagrange multipliers, Convex sets and functions, Elements of Gradient search algorithms: Steepest descent, Newton and Jacobi algorithms, Least Squares method, Application: The Markowitz Model, and Overview of constrained optimization, Hill climbing, Single variable search. Recap of Unit I.

**Unit II: Infinite Series and Introduction to Complex Analysis 13 Lecture Hours**

Sequence and series, Convergence tests: p-series, Comparison, Ratio and root test, Alternating series. Complex number system, Euler’s formula, Functions of a complex variable, Hyperbolic functions, Limit and Continuity, Derivative and Analytic functions, Holomorphic functions, Cauchy-Riemann equations, Harmonic functions, Line integral and independence of path, Cauchy’s theorem, Cauchy’s integral formula, Zeros and singularities of a function, Power series: Taylor’s and Laurent’s series. Some applications. Recap of Unit II.

**Unit III: Introduction to Special Functions 07 Lecture Hours**

Introduction to Power series method, Legendre’s equation and Legendre polynomials, Bessel’s equation and Bessel functions. Application of Bessel functions: CV Raman’s model of Indian drums. Recap of Unit III.

**Unit IV: Integral Transforms 15 Lecture Hours**

Laplace Transform and its properties, Shifting Theorems, Laplace Transform of derivatives, integrals, and periodic functions, Heaviside and Dirac Delta Functions. Inverse Laplace transforms, Convolution, Solutions of differential equations using Laplace transforms. Fourier series and applications, Dirichlet’s condition, Fourier Transforms, Fourier sine and cosine transforms, Properties of Fourier Transforms, Fast Fourier Transform, Inverse Fourier transforms. Recap of Unit IV.

**Unit V: Introduction to PDEs and Applications 11 Lecture Hours**

Introduction to Partial differential equations (PDE) and real-world applications, Classification of PDEs: Elliptic, Hyperbolic, Parabolic, Solution of homogeneous and non-homogeneous linear PDEs, Method of separation of variables using Fourier series, Solution of Heat conduction or Diffusion equation, Connection between diffusion and randomness, Wave Equation, Laplace Equation, and Poisson Equation. Some applications: Air pollution, Traffic model. Recap of Unit V.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. E. Kreyszig, "Advanced Engineering Mathematics", 9th Edition, Wiley Publications, 2016. 2. G. James, and P. P. Dyke, "Advanced Modern Engineering Mathematics", 5th Edition, Pearson Education. 2018. 3. J. P. Corriou, "Numerical Methods and Optimization: Theory and Practice for Engineers", Springer, 2021. |
| **Reference books** | 1. D. G. Zill, and P. D. Shanahan, "Complex analysis", 3rd Edition, Jones & Bartlett Learning, 2015. 2. W. A. Strauss, "Partial Differential Equations - An Introduction", 2nd Edition, John Wiley & Sons Inc., 2008. 3. L. Burstein, "PDE Toolbox Primer for Engineering Applications with MATLAB® Basics", CRC Press, 2022. 4. I. Goodfellow,Y. Bengio, and A. Courville, "Deep learning", MIT Press., 2016. (Sections 4.3, 4.4, and 4.5) 5. S. Chandra, Jayadeva, and A. Mehra, "Numerical optimization with applications", Narosa, 2009. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SSEN0102** | **Environment Sustainability and Climate Change (Living Lab)** | | 0 | 0 | 2 | 2 |
| **Total Units to be Covered: 07** | | **Total Contact Hours: 30** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the concept of Living Labs and their application in the environmental sustainability.
2. Develop a critical understanding of the nature, cause and impact of human activities on the environment.
3. Apply design thinking and innovative principles to develop sustainable solutions.
4. Evaluate and address legal, policy and ethical consideration in environmental research.

**Course Outcomes**

1. Gained practical skills in stakeholder engagement, environmental data collection and analysis.
2. Develop expertise in designing and managing Living Lab for environmental sustainability and climate action.
3. Acquired hands on experience with environmental monitoring tools and technologies.
4. Enhance the ability to think critically and creatively in developing sustainable solutions.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | 2 | 1 | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | 3 | 1 | **-** | 1 | 2 | **-** | **-** | 1 | **-** | **-** | 2 | **-** | **-** | **-** | **-** |
| **CO 4** | 2 | 2 | **-** | 1 | 1 | **-** |  | 1 | **-** | **-** | 1 | **-** | **-** | **-** | **-** |
| **Average** | 2.25 | 1 | **-** | .5 | **-** | **-** | **-** | .5 | **-** | **-** | .75 | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

1. A two-credit course in practicum or lab work, community engagement and services, and field work in a semester means two-hour engagement per week. In a semester of 15 weeks duration, a one credit practicum in a course is equivalent to 30 hours of engagement.
2. Case Studies and Field Work

The students are expected to be engaged in some of the following or similar identified activities:

Discussion on one national and one international case study related to the environment and sustainable development.

Examples: Bhopal Gas Tragedy, Chipko Movement, Narmada Valley Projects, National Park, Sanctuaries, Biosphere Reserve, London Smog 1952, Air Pollution in Delhi, Case studies on Current Environmental Issues, Oil Spills – Deep Water Horizon Oil Spill, BP Oil Spill etc.

1. **Field Visit**

Field visits to identify local/regional environmental issues, make observations including data collection and prepare a brief report.

1. Campus Environmental Management.

Campus environmental management activities such as solid waste disposal, water management, and sewage treatment.

Group Project: Students are required to submit group projects on various topics related to environmental pollution, climate change, biodiversity, natural resource and sustainable development.

**Broadly, Living Lab may falls in one of seven thrust areas:**

1. **Indigenous technology and Traditional Ecological Knowledge (TEK)**

The project aims to document, preserve and revitalize Traditional Ecological Knowledge (TEK) held by indigenous communities. TEK encompasses the deep understanding of local ecosystems, sustainable solution resource management practices and cultural connection to environment. This project emphasizes a collaborative approach involving indigenous elders, community members and researcher. The project not only respects and preserves the rich cultural heritage of indigenous communities but also harnesses their valuable ecological knowledge to address contemporary environmental challenges and promote sustainable practices. It emphasizes the importance of community-driven conservation efforts and the recognition of TEK as a valuable source of ecological wisdom.

1. **Climate change and its impact on Bird Migration**

In recent years, climate change has been affecting the migration patterns of many bird species worldwide. This project aims to study and mitigate the impact of climate change on avian migration and contribute to conservation efforts. This project not only addresses the ecological impact of climate change but also contributes to the conservation of bird species that play vital roles in ecosystem health and biodiversity. It emphasizes the importance of interdisciplinary collaboration and community engagement in tackling climate-related ecological challenges. This project will focus on developing targeted conservation strategies to mitigate the impact of climate change on bird population and help in enhancing collaboration among scientists, conservationists, and local communities for bird conservation.

1. **Sustainable Communities**

How can co-production and social learning with stakeholder communities help us understand how climate action can be implemented ‘on the ground’?

The ‘living lab’ offer for active engagement with a diverse student body and neighborhood groups. It reflects the wider academic recognition that universities are significant economic, social and environmental catalysts for cities and regions, offering the potential for change at a spatial scale that connects the local with the global. Project activities to empower the local community-based people to enhance their lifestyle by doing activities. For example, we can do a few projects like utilization of Himalayan biomass for various uses. Our students can give this training and awareness program to localities.

We can work on the SMART village project by following the SDG goals given by United Nations.

Identification and selection of such communities who have some native or ancestral knowledge. For example, one farmer in Kerala has huge seed bank from the very old time (more than 200 Years).

Projects in this area will explore how meaningful policy change can be driven in expanding circles from the level of university communities to the cities, states and nations they are embedded in.

1. **Ecology, Conservation, and Climate Change**

Project within this domain will investigate the ecological characteristics of ecosystem undergoing degradation, examine the dynamics of shifts in parasite ecology and explore the enduring adaptation in hosts, parasites and explore the evolutionary adaptation in host, parasites and wildlife influenced by climate fluctuation and various environmental stressors.

UPES can work on the preservation of untouched Himalayan flora and fauna and propose one flora and fauna bank. This may be followed by the several awareness program for the locals by our staffs and students. The primary aim of projects in this area will be to establish and sustain long-term studies of how climate change impacts ecological aspects of our natural world.

1. **One Health**

This project is centered around the concept of "One Health," which is an approach that recognizes the interconnectedness of human health, animal health, and the health of the environment. It aims to address the broader context of ecological health, acknowledging that various factors such as climate change, habitat alterations, and biodiversity loss play a significant role in shaping the overall health of ecosystems and, consequently, human health. In essence, this project seeks to broaden our understanding of health by looking beyond the human dimension and recognizing the intricate web of relationships that connect human health to animal health and the environment.

One of the primary concerns is to examine the potential sources of new zoonotic diseases. Zoonotic diseases are those that can be transmitted from animals to humans, and understanding their origins is crucial for preventing future outbreaks.

1. **Climate and society**

Climate action will require disruptive transformations for society, but how can that ‘transformational intent’ be developed? The digital revolution and data science show how rapid transformation can happen, history offers perspectives on such transformations in the past while the business world illuminates how rapid change continues to occur in the commercial and financial sector.

This group of projects will study the interface between climate, sustainable living, and the forces that drive society and societal change.

1. **Communities Based Water Testing Kits and Soil Testing Kits and promotion of low cost water purification technologies**

Identify water borne disease in surrounding villages by conducting field surveys and sensitize local communities about water borne diseases and suggest low-cost water treatment methods.

Training local people for water and soil testing.

Sampling and analysis of drinking water

**Total lecture Hours 30**

# References\*

|  |  |  |
| --- | --- | --- |
| **Textbooks** |  |  |
| **Reference books** |  |  |
| **Web Resources** |  |  |
| **Journals** |  |  |
| **MOOCs, online courses** |  |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |
| --- | --- |
| **Components** | **Continuous Assessment** |
| Weightage (%) | 100% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SFLS0002** | **Time & Priority Management** | |  |  |  | **2** |
| **Total Units to be Covered: 9** | | **Total Contact Hours: 30** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**SEMESTER III**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG2060** | **Operating Systems** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** | **Linux Lab -** **CSEG1126** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

To equip students with a deep understanding of operating system design and implementation, enabling them to analyze, evaluate, and apply concepts such as process management, concurrency, memory management, and storage in real-world scenarios.

**Course Outcomes**

CO1: Demonstrate a comprehensive understanding of operating systems.

CO2: Evaluate and analyze process and thread scheduling techniques, discerning their benefits and challenges.

CO3: Demonstrate an understanding of inter-process communication (IPC) mechanisms, process synchronization and deadlocks.

CO4: Evaluate and analyze memory and storage management techniques.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | 1 | 1 | 1 | 1 | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 3 | **-** | **-** |
| **CO 2** | 1 | 2 | 1 | 3 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 3 | **-** | **-** |
| **CO 3** | 2 | 2 | 1 | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 3 | **-** | **-** |
| **CO 4** | 2 | 2 | 1 | 2 | 2 | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 3 | **-** | **-** |
| **Average** | 1.5 | 1.75 | 1 | 2 | 2.25 | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 3 | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Operating System 6 Lecture Hours**

Computer Hardware Review; Computer System; Introduction to Operating System: Definition, Operating System view, History, Types of Operating Functions of Operating System, Services of Operating System, Computing Environments, Virtualization and Containerization, Operating System Structures, Operating System Operations, System boot. System Calls, Types of System Calls (Windows and Unix System Calls examples), Open Source Operating Systems

**Unit II: Process and Thread Management 8 Lecture Hours**

Process: Program and Process concept, Process in memory, Process Control Block, Process States, Process Context Switching, Process Scheduling Queues, Process Schedulers, Process Context Switching, Process Scheduling Criteria, Process Scheduling: Non preemptive and Preemptive Schedulers, FCFS, Shortest Job First, Shortest Remaining Time First, Non Preemptive Priority scheduling and Preemptive Scheduling, Priority Round Robin, Multilevel Queue, Multiple Feedback Queue, Real Time scheduling : Rate Monotonic, Earliest Deadline First; Operations on processes: Creation and Termination

Threads: Threads and its benefits, Multi-threading models, Kernel Level thread, user level thread and hybrid threads, Thread Scheduling: Content Scope, Pthread Scheduling, Threading Issues

Case study: Process Management in Linux

**Unit III: Inter Process Communication and Synchronization 9 Lecture Hours**

Inter Process Communication (IPC), IPC mechanisms: Shared Memory and Message Passing (Shared Memory, Pipes and Named pipes in Linux), Critical Section Problem, Race Condition, Producer Consumer Problem, Solution to Critical section Problem: Hardware and Software Solutions, Software Solutions: Semaphores: Counting semaphore, Binary semaphore, Monitors, Algorithm 1, Algorithm 2, Algorithm 3/Peterson Solution, Bakery Algorithm, Classic process synchronization problems (case studies).

**Unit IV: Deadlock Handling 6 Lecture Hours**

Deadlock, Deadlock characterization: Necessary Conditions for Deadlock, Resource Allocation Graph; Methods for Handling Deadlocks: Deadlock Prevention, Deadlock Avoidance: Safe State, Resource Allocation Graph Algorithm, Bankers Algorithm; Deadlock Detection; Recovery from deadlock: process Termination and Resource Preemption

**Unit V: Memory Management 9 Lecture Hours**

Memory protection, Address binding, Logical versus Physical Address Space, Dynamic Loading, Dynamic Linking, Swapping, Memory Management Strategies: Contiguous and Non-Contiguous; Contiguous memory management: static and dynamic: First Fit, Best Fit, Worst Fit, Buddy System, Internal Fragmentation, External Fragmentation, Compaction, Non Contiguous memory management: Paging, Paging Hardware Support, Structure of Page Table: Hierarchical paging, Hashed page tables and Inverted page tables, Virtual memory, Demand Paging, Page Fault, handling of Page Fault, Page Replacement, Page Replacement Algorithms, Belady’s anamoly, Allocation of frames, Thrashing, Segmentation: concept, Segmentation hardware, Segmentation with Paging

Case Study: Memory Management in Linux

**Unit VI: Storage Management 7 Lecture Hours**

File concepts, File system structure, File attributes, File operations, File types, File access method, File system mounting, Directory, Different logical structure of directories, Disk structure, Disk allocation methods: contiguous, linked and indexed, Free space management, Disk scheduling algorithms.

Case Studies: File System in Linux and Windows

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, "Operating System Concepts", 10th Edition, John Wiley & Sons, 2018. 2. Andrew S. Tanenbaum and Herbert Bos, "Modern Operating Systems", 4th Edition, Pearson, 2021. 3. William Stallings, "Operating Systems: Internals and Design Principles", 9th Edition, Pearson, 2021. |
| **Reference books** | 1. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, "Operating Systems: Three Easy Pieces", 1st Edition, CreateSpace Independent Publishing Platform, 2018. 2. J. Archer Harris, "Schaum's Outline of Operating Systems", 1st Edition, McGraw-Hill Education, 2002. 3. Garry J Nutt, "Operating System – A modern perspective", 2nd Edition, Addison Wesley, 2002. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG2065** | **Data Communication and Networks** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The objectives of this course are as follows:

1. Understand the basic components and functions of computer networks, including network topologies, protocols, and networking devices.
2. Understand need of layered architecture and differentiate OSI and TCP/IP
3. Gain an understanding of error and flow control techniques on communication channels.
4. Explore routing algorithms and its application.
5. Get a brief idea about network analysis tools (Wireshark, NMAP).

**Course Outcomes**

The outcomes of this course are as follows:

**CO1:** Evaluate network devices functionality and network command significance.

**CO2:** Evaluate and address problems of error control, flow control, and channel access.

**CO3:** Analyze and adopt fundamental workings of routing algorithms.

**CO4:** Create solutions for recent challenges in large-scale networks.

**CO5:** Apply knowledge of network traffic analysis tool to investigate network activities.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | 2 | 3 | **-** | **-** | **-** | **-** | 2 | **-** | **-** | 3 | **-** | **-** | **-** |
| **CO 2** | **-** | 2 | 3 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 2 | **-** | **-** |
| **CO 3** | 2 | 3 | 3 | 3 | 2 | 2 | **-** | **-** | 2 | **-** | **-** | 3 | 3 | 2 | **-** |
| **CO4** | 3 | 3 | 3 | 3 | 2 | **-** | **-** | 2 | 2 | **-** | 2 | 3 | 2 | 3 | **-** |
| **CO5** | 2 | 3 | 3 | 3 | 3 | 3 | **-** | 2 | 2 | 2 | 3 | 3 | 3 | 3 | **-** |
| **Average** | 1.8 | 2.6 | 2.8 | 2.8 | 1.4 | 1 | **-** | 0.8 | 1.6 | 0.4 | 1 | 3 | 2 | 1.6 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Basic of Computer Networking and Technologies 9 Lecture Hours**

Introduction to Computer Networking Concepts: Layered Network Protocol Architectures (OSI, TCP/IP); LAN, WAN, MAN, PAN, LAN Topologies; Connectivity Devices and Cable Types; Ethernet, Gigabit Ethernet (GbE); Circuit Switching, Message Switching, and Packet Switching; WiMAX, 5G and beyond, Cellular Technology, Communication Channels and performance metrics.

**Unit II: Data Link Layer - Part 1 9 Lecture Hours**

Logical Link Control (LLC) sub-layer: Framing, Data Communication Character Codes, Error Control: Error Detection (Redundancy Checking: VRC, Checksum, LRC, CRC); Retransmission, Error Correction: Forward Error Correction (Hamming Code), Character Synchronization, Reliable transmission and Automatic Repeat Request (ARQ) protocols including Stop-and-Wait, Go-back-N, Selective Repeat; Performance analysis of ARQ protocols; Example protocols such as HDLC and PPP.

**Unit III: Data Link Layer - Part 2 8 Lecture Hours**

Medium Access Control (MAC) sub-layer: Channel Allocation Problems, Multiple Access Protocols and Types: TDMA, FDMA, CSMA, CSMA/CD, CSMA/CA protocols; Hidden Node and Exposed Node Problems, Performance analysis; Shared and Switched Ethernet; IEEE Standards 802.3 & 802.11, 10-Gigabit Ethernet.

**Unit IV: Network Layer 10 Lecture Hours**

Network Layer Design Issues, Network Address Translation, Internet Protocol (IP): IPv4 and IPv6 addressing; IP Addressing Techniques: Classful Addressing, Classless Addressing, Network and Host Identification, Loopback Address, Broadcast Address, Address Masking; Networks and Subnetworks: Subnetting, Subnet Mask, Supernetting; Network-Layer Protocols: ARP, RARP, IP datagram; Internetworking: Routing and Routing protocols (distance-vector and link-state); Interior and Exterior Gateway Protocol concepts; Routing Algorithms including Dijkstra's algorithm and distributed Bellman-Ford algorithm; Example protocols: OSPF, RIP, BGP, Encapsulation and Tunneling, Congestion Control, Quality of Service, Introduction of Wireshark Tool.

**Unit V: Transport Layer 9 Lecture Hours**

Introduction and Transport-Layer Services, Port Address, Socket Address; Internet Transport Protocols: UDP, Introduction to UDP, Remote Procedure call, Real-time Transport Protocols; Internet Transport Protocols: TCP, service model, TCP protocol, TCP segment header, TCP Connection establishment, TCP Connection Release, TCP Connection management modeling, TCP sliding window, TCP Timer management, TCP Congestion control; Performance Issues: Performance problems in computer networks, Network Performance Management, Host Design for fast networks; Fast segment processing, Header compression, protocols for long Fat networks; Virtual Private Network (VPN); Introduction of Nmap Tool.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. James F. Kurose, and Keith W.Ross, "Computer Networking : A Top-Down Approach", 8th Edition, Pearson, 2022. 2. W. Tomasi, "Introduction to data communications and networking", 5th edition, Prentice-Hall, Inc., 2008. |
| **Reference books** | 1. Walter Goralski, "The illustrated network: how TCP/IP works in a modern network", 2nd Edition, Morgan Kaufmann, 2017. 2. Andrew S. Tanenbaum, "Computer Networks", 5th Edition, Pearson Education, 2023. 3. L. L. Peterson, and B. S. Davie, "Computer networks: a systems approach", 6th Edition, Elsevier, 2020. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Data communication and Networks Lab**

**List of Experiments**

**Experiment 1:** Familiarization with networking devices. **(CO1)**

**Experiment 2:** Write a program for bit stuffing and de-stuffing in a bit stream. **(CO2)**

**Experiment 3:** Write a program for CRC and Hamming Code. **(CO2)**

**Experiment 4:** Familiarization with Network IP, subnetting and supernetting. **(CO3)**

**Experiment 5:** Familiarization of basic network command and network configuration commands. **(CO1, CO5)**

**Experiment 6:** Set up a network topology in Cisco Packet Tracer (Ring, Bus, Star, Mesh etc.) **(CO4, CO5)**

**Experiment 7:** Set up network topology in two and more than two routers. **(CO4, CO5)**

**Experiment 8:** Distance vector routing protocol **(CO3)**

**Experiment 9:** Link-state vector routing protocol **(CO3)**

**Experiment 10:** Familiarization with network monitoring tools (NMAP and Wireshark) **(CO5)**

**Experiment 11:** Capture network trafficusing Wireshark. **(CO5)**

**Experiment 12:** Analyzing network traffic using Wireshark. **(CO5)**

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. James F. Kurose, and Keith W.Ross, "Computer Networking : A Top-Down Approach", 8th Edition, Pearson, 2022. 2. Andrew S. Tanenbaum, "Computer Networks", 5th Edition, Pearson Education, 2023. |
| **Reference books** | 1. W. Tomasi, "Introduction to data communications and networking", 5th edition, Prentice-Hall, Inc., 2008. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG1032** | **Computer Organization and Architecture** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

To equip students with the necessary knowledge and skills to comprehend, analyze, and design digital computer systems, ensuring they can effectively develop and optimize software applications and systems.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Analyze the components and organization of digital computers.

**CO2:** Apply knowledge of instruction codes, instruction formats, and addressing modes to analyze and design computer instructions in different CPU architectures.

**CO3:** Examine the design and organization of control units in digital computers to comprehend their role in executing instructions and managing system operations.

**CO4:** Analyze the organization and performance implications of memory units and input –output systems in digital computer systems.

**CO5:** Assess the benefits and challenges of pipelining in computer architecture on system performance and throughput.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 |  | 2 | **-** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | 3 | 2 | **-** | **-** |
| **CO 2** | 1 | 1 | 2 | **-** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | 3 | 2 | **-** | **-** |
| **CO 3** | 1 | 1 | 2 | **-** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | 3 | 2 | **-** | **-** |
| **CO 4** | 1 |  | 2 | **-** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | 3 | 2 | **-** | **-** |
| **CO 5** | 1 | 1 | 2 | **-** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | 3 | 2 | **-** | **-** |
| **Average** | 1 | 0.6 | 2 | **-** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | 3 | 2 | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Digital Computers 8 Lecture Hours**

Introduction; Block Diagram for Digital Computers: CPU (Registers, ALU, Clock, Control unit), Memory, I/O subsystems, Common Bus System (External and Internal Bus: Address Bus, Data Bus and Control Bus); Computer Organization; Computer Architecture; Introduction to Vonn Neumann and Harvard Architecture, Data representation: Number System, r complement and r-1 complement arithmetic, Unsigned and Signed number representation, Big Endian and Little Endian, Signed Arithmetic- Addition, Subtraction, Multiplication (Booth Algorithm), Division, Fixed and Floating point representation. Register Transfer Language (RTL) and Micro operations (Arithmetic, Logical and Shift micro operations), Arithmetic Logic and Shift unit (ALU).

**Unit II: Basic Computer Organization and Design 7 Lecture Hours**

Instruction Codes; Instruction Format (Three-Address Instructions, Two-Address Instructions, One-Address Instructions, Zero-Address Instruction); Computer Instructions, Registers (General Purpose and Special Purpose Registers); General Register Organization, Stack organization, Types of Instructions (Memory Reference, Register Reference and Input Output Instructions); Addressing Modes and its types; Instruction cycle, Interrupt cycle;

*Case Study: Some common CPU architectures (Intel IA-32 Architecture, ARM),*

**Unit III: Control Unit Organization 6 Lecture Hours**

Hardwired Control Unit and Timing Signals, Microprogrammed control unit: control memory, Address sequencing, Microprogram Example, Designing of microprogrammed control unit. Comparison of hardwired and microprogrammed control units, RISC and CISC Processors;

*Case Study: Designing a hypothetical processor with minimum number of instructions so that it can perform basic arithmetical and logical operations.*

**Unit IV: Memory Organization 8 Lecture Hours**

Memory hierarchy; Different types of memory: Primary (RAM-Static and Dynamic, ROM-EPROM,EEPROM, Cache-Level 1, Level 2 and Level 3) and Secondary/Auxiliary Memory (Magnetic Disk), Introduction to emerging in-situ memory technologies- ReRAM, PCM, STTRAM; Main Memory: RAM and ROM Chips, Memory Address Map, Memory Connection to CPU; Associative Memory. Cache Memory: Principle of Locality, Cache mapping techniques; Performance considerations: Hit Rate and Miss Penalty, cache coherence, cache read and write policy, caches on the Processor Chip.

**Unit V: Input Output Organization 8 Lecture Hours**

Peripheral Devices; I/O interface; I/O bus and interface modules; I/O Bus vs Memory Bus, Interrupts and Types of Interrupts. Modes of data transfer: Programmed, Interrupt-initiated, Direct Memory Access (DMA), Priority Interrupt, Input Output processor.

**Unit VI: Pipelining 8 Lecture Hours**

Multiprogramming, Multiprocessing, Single instruction single data stream (SISD); Single instruction multiple data stream (SIMD); Multiple instruction single data stream (MISD); Multiple instruction multiple data stream (MIMD), Multiprocessors: Shared memory and distributed memory, Parallel processing: Pipeline processing, Vector processing, Array processors. Pipelining: Arithmetic Pipeline, Instruction Pipeline: Example: Four-Segment Instruction Pipeline. Pipelining Conflicts: Resource conflicts, Data dependency and Branch difficulties, Pipeline Conflicts Handling techniques: Throughput and Speed; RISC pipeline;

*Case study: Pipelining in CISC Processors, Pipelining in ColdFire Processors, and Pipelining in Intel Processors, Designing pipeline architecture for 2, 3, 4 stage pipeline.*

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. M. M. Mano, "Computer System Architecture", Revised 3rd Edition, Pearson Education, 2017. 2. Carl Hamacher, Zvonko Vranesic , Safwat Zaky, and Naraig Manjikian, "Computer Organization and Embedded Systems", 6th Edition, McGraw Hill, Standard Edition, 2023. 3. David A. Patterson, and John L. Hennessy, "Computer Organization and Design MIPS Edition: The Hardware/Software Interface", 5th Edition, The Morgan Kaufmann Series in Computer Architecture and Design, Morgan Kaufmann, 2020. |
| **Reference books** | 1. John P. Hayes, "Computer Architecture and Organization", 3rd Edition, McGraw-Hill Education, 2017. 2. William Stallings, "Computer Organization and Architecture: Designing for Performance", 11th Edition, Pearson, 2022. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG3053** | **Design and Analysis of Algorithms** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | Data structures and algorithms -  CSEG1043 | | **Syllabus version: 1.0** | | | |

**Course Objectives:`**

The objectives of this course are as follows:

1. Define the fundamental concepts, definitions and terminologies related to algorithms, data structures, and algorithm analysis.
2. Understand the principles behind various algorithm design techniques.
3. Apply algorithms analysis techniques to evaluate the efficiency and asymptotic performance of algorithms in terms of time and space complexity.
4. Analyse and compare different algorithmic solutions for the same problem, considering their efficiency, correctness, and suitability for specific scenarios.

**Course Outcomes**

The outcomes of this course are as follows:

**CO1.** Demonstrate a solid understanding of fundamental concepts, terminologies, and principles related to algorithms, data structures, and algorithm analysis.

**CO2.** Apply algorithmic design techniques to solve real-world problems.

**CO3.** Compare and contrast multiple algorithmic approaches for the same problem, considering their efficiency, correctness, and practicality.

**CO4.** Select problem-solving strategies and algorithmic thinking to tackle new and challenging problem domains and classify the algorithms in different classes.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 3 | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 1 | **-** |
| **CO 2** | 3 | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 1 | **-** |
| **CO 3** | 2 | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 1 | **-** |
| **CO 4** | 2 | 3 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 1 | **-** |
| **Average** | 2.5 | 2.75 | 1.75 | 1.5 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 1 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Algorithms 9 Lecture Hours**

Algorithm, Characteristics of algorithm, Distinct area of study of algorithm, Different algorithm design techniques, Algorithm analysis, Growth of an algorithm, Asymptotic notations, Recurrence relation, Solving recurrence relation by iteration, substitution, recursion tree, master theorem method.

**Unit II: Algorithm design paradigm: Divide and Conquer 7 Lecture Hours**

The divide and conquer paradigm, Analysis of Binary search, Merge sort, Quick sort, Strassen Method of Matrix Multiplication, Maximum subarray problem, Powering number, Celebrity problem.

**Unit III: Algorithm design paradigm: Greedy Method 7 Lecture Hours**

Greedy approach design paradigm, Knapsack problem, Activity selection problem, Huffman encoding, Interval partitioning problem, Dijkastra algorithm for single source shortest path problem, Prim’s and Kruskal algorithm for finding minimum cost spanning tree.

**Unit IV: Algorithm design paradigm: Dynamic Programming 7 Lecture Hours**

Dynamic programming design paradigm, 0/1 Knapsack problem, Matrix chain multiplication problem, longest common subsequence problem, Optimal binary search problem, Bellman ford algorithm for single source shortest path problem, Travelling salesman problem, Difference between divide and conquer, greedy and dynamic programming algorithm design approach, Floyd warshall algorithm for all pair shortest path problem

**Unit V: Algorithm design paradigm: Backtracking 7 Lecture Hours and Branch & Bound**

Introduction to backtracking and branch & bound approach, backtracking based problems: N Queen problem, Sum of Subset problem, 0/1 Knapsack problem, Branch & Bound based problems: FIFO, LIFO, & LC branch & bound, 0/1 Knapsack problem, Travelling salesperson problem.

**Unit VI: Maximum Flow and String-Matching Problems 8 Lecture Hours**

Flow networks: Ford- Fulkerson method, Maximum bipartite matching, Modulo Representation of integers/polynomials: Chinese Remainder Theorem, String Matching: The naive string-matching algorithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm, Different classes of problems: P, NP, NP Complete, NP Hard, reducibility property

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. T. H. Cormen, C. F. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", 4th Edition, MIT Press, 2022. 2. 2. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Galgotia Publication, 2010. |
| **Reference books** | 1. 1. Jon Kleinberg, and Eva Tardos, "Algorithm Design", Addison Wesley, 2005. 2. 2. A. V. Aho, J. Hopcroft, and J. D. Ullman, "The Design and Analysis of Algorithms", Addison-Wesley, 2002. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Design and Analysis of Algorithms Lab**

**Syllabus**

**List of Experiments**

**Experiment 1-4: Divide and Conquer Approach**

* Implement the iterative and recursive Binary search tree and compare their performance.
* Implement divide and conquer based merge sort and quick sort algorithms and compare their performance for the same set of elements.
* Compare the performance of Strassen method of matrix multiplication with traditional way of matrix multiplication.

**Experiment 5-9: Greedy & Dynamic Programming Approach**

* Implement the activity selection problem to get a clear understanding of greedy approach.
* Get a detailed insight of dynamic programming approach by the implementation of Matrix Chain Multiplication problem and see the impact of parenthesis positioning on time requirements for matrix multiplication.
* Compare the performance of Dijkastra and Bellman ford algorithm for the single source shortest path problem.
* Through 0/1 Knapsack problem, analyze the greedy and dynamic programming approach for the same dataset.

**Experiment 10-13: Backtracking and Branch & Bound Approach**

* Implement the sum of subset and N Queen problem.
* Compare the Backtracking and Branch & Bound Approach by the implementation of 0/1 Knapsack problem. Also compare the performance with dynamic programming approach.

**Experiment 14-15: String Matching Problems**

* Compare the performance of Rabin-Karp, Knuth-Morris-Pratt and naive string-matching algorithms.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. T. H. Cormen, C. F. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", 4th Edition, MIT Press, 2022. 2. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Galgotia Publication, 2010. |
| **Reference books** | 1. Jon Kleinberg, and Eva Tardos, "Algorithm Design", Addison Wesley, 2005. 2. A. V. Aho, J. Hopcroft, and J. D. Ullman, "The Design and Analysis of Algorithms", Addison-Wesley, 2002. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SLLS2001** | **Social Internship** | | **0** | **0** | **0** | **0** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Exploratory-1** | | 3 | 0 | 0 | 3 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SFLS0003** | **Communication Skills (Oral)** | |  |  |  | **2** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**SEMESTER IV**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG1044** | **Object Oriented Programming** | | 3 | 0 | 1 | 4 |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | Programming in C - CSEG1025 | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the need for OOPs and develop Java programs with object-oriented features.
2. Learn the concepts of JDBC and develop standalone application with GUI Panel.
3. Design & implement Java applications for real world scenarios.

**Course Outcomes**

CO1. Understand Object Oriented Programming concepts and architecture of Java.

CO2. Analyze and model the real-world entity using Java programming language.

CO3. Develop packages with Generics and Implement Interfaces with Exception handling.

CO4. Create Stand-alone Java applications using GUI swings and JDBC.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |
| **CO 2** | **-** | 3 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |
| **CO 3** | **-** | 3 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |
| **CO 4** | **-** | **-** | **-** | 2 | **-** | **-** | 1 | **-** | 2 | 2 | **-** | **-** | 2 | 3 | **-** |
| **Average** | .25 | 1.5 | 1.5 | .5 | **-** | **-** | .25 | **-** | .5 | 1 | **-** | **-** | 2 | 3 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to OOPs 5 Lecture Hours**

Object Oriented Programming History and Evolution, Object Oriented Programming Principles, Features of Java, Input Output Statements, Comment Line Arguments, Data Types, Variables, Operators, Program Control Statements, Arrays, Type of Arrays, Strings.

**Unit II: Classes, Inheritance, Packages and Interfaces 8 Lecture Hours**

Class Fundamentals, Objects, Constructors, Garbage Collection, this Keyword, Java’s Access Modifiers, Method Overloading, static Keyword, Inheritance, Types of Inheritance, super to Access Superclass Members, Method Overriding, Abstract Classes, Using final, Packages and Interfaces, Build-in Interface, User defined Interfaces.

**Unit III: Nested Classes, Exceptions, Multithreading & IO Streams 8 Lecture Hours**

Nested Classes, Types of Nested Classes, Exception Handling, Exception Handlers, Concurrent Programming, The Thread Class and Runnable Interface, Thread Priorities, Synchronization, Java’s I/O Streams, Byte Streams and Character Streams, FileWriter, FileReader.

**Unit IV: Generics, Lambdas, GUI Swing & Database Connectivity 8 Lecture Hours**

Generics Fundamentals, Generic Class, Generic Methods, Lambdas, Functional Interfaces, Swing, Components and Containers, Layout Managers, Swing Event Handling, Event Listeners, Event Classes and Listener Interfaces, Swing Controls, Database Connectivity, Statement, Prepared Statement, CallableStatement, Resultset. Persistent Data.

**Unit V: Collections and Wrapper Class 6 Lecture Hours**

Collections, Iteration, Collection Interface, Set and SortedSet, List, Map and SortedMap, Wrapped Collections and Collections Class, Wrapper classes and loading classes.

**Unit VI: Capstone Project 10 Lecture Hours**

Create Standalone Java Project, Designing of UML and database diagrams, GUI Panel development using swing, Establish connection with Database and Panel. Source Code Management and Collaboration using Git/GitHub. Unit Testing using JUnit, Integration Testing, Build and Artifactory Management.

**Total lecture Hours 45**

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. Herbert Schildt, "Java: A Beginner's Guide", 9th Edition, McGraw-Hill Education, 2022. 2. 2. Allen B. Downey and Chris Mayeld, "Think Java: How to Think Like a Computer Scientist", 2nd Edition, O'Reilly Media Publishers, 2020. |
| **Reference books** | 1. 1. Herbert Schildt, "Java: The Complete Reference", 12th Edition, McGraw Hill Publisher, 2022. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Object Oriented Programming Lab**

**List of Experiments**

|  |  |
| --- | --- |
|  |  |
| **Experiment 1** | Introduction to Java Environment |
| **Experiment 2** | Basic Java Programming |
| **Experiment 3** | Basic Java Programming |
| **Experiment 4** | Inheritance |
| **Experiment 5** | Interface |
| **Experiment 6** | Package |
| **Experiment 7** | Exceptions |
| **Experiment 8** | Strings Handling and Wrapper Class |
| **Experiment 9** | Threads and Collections |
| **Experiment 10** | JDBC |
| **Experiment 11** | Servlets |

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. Ken Arnold, and James Gosling, "The Java Programming Language", 3rd Edition, Pearson, 2018. 2. 2. Khalid Mughal, "A premier guide to SCJP", 3rd Edition, Pearson. 3. 3. Bruce Ackel, "Thinking in Java", 3rd Edition, Pearson. 4. 4. Video resources http://www.youtube.com and blackboard. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG2072** | **Database Management Systems** | | **3** | **0** | **2** | **5** |
| **Total Units to be Covered: 7** | | **Total Contact Hours: 105** | | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the concept of DBMS and ER Modelling.

2. To explain normalization, Query optimization and relational algebra.

3. To apply concurrency control, recovery, security and indexing for real time data.

**Course Outcomes**

**CO 1**  Understand the foundational concepts of data models, schema design, and relational databases to effectively manage and query data.

**CO 2** Learn to design efficient and normalized databases, apply entity-relationship modeling, and optimize schema structures.

**CO 3**  Acquire skills in database security, user access control, backup and recovery, and performance tuning to ensure robust database management.

**CO 4** Develop database application design and its implementation including integrity constraints, transaction management and concurrent control algorithms.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |
| **CO 2** | **-** | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |
| **CO 3** | **-** | 3 | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |
| **CO 4** | **-** | 3 | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |
| **Average** | **-** | 3 | 2 | 2.5 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Databases 7 Lecture Hours**

Introduction to Database, Database users, characteristics and advantages of the database, Database systems, Concepts and architecture-Data models, schemas & instances, Three-Schema architecture & data independence, database languages & interfaces, Centralized and Client/Server Architecture of DBMS.

**Unit II: Data Modelling 8 Lecture Hours**

Data Modelling, Using the Entity Relationship (ER) Model, The Enhanced Entity-Relationship (EER) Model: - Entity Set, attributes and their types, Relationship Constraints (including Participation constraints and cardinality ratio), ER Diagrams, constraints and design issues, Reduction of ER and EER diagrams to relational schemas. UML Class Diagrams.

**Unit III: Relational Database Design and Normalization 8 Lecture Hours**

Relational model Concepts, constraints, and relational database schemas, transactions, and dealing with constraint violations, DBMS Keys, Relational Algebra, Ary relational operations, Binary relational operations, and relational algebra operations from set Theory, Relational Calculus, and implementation in SQL.

Informal Design guideline for relational Schemas, Functional Dependencies, Normal forms based on primary keys (1NF, 2NF, 3NF & BCNF), lossless join and dependency-preserving decomposition, Multivalued dependencies (4NF, 5NF), and domain key normal form.

SQL- Queries, Constraints, Form of SQL query, UNION, INTERSECT, and EXCEPT, Nested queries, Aggregate Operators, Null values, Complex Integrity constraints in SQL, and triggers.

**Unit IV: DBMS Architecture, Query Processing and Optimization 7 Lecture Hours**

DBMS Instance, DBMS Internal Memory Structure, Background Processes, Data Types, Roles & Privileges, Introduction to Query Processing, Translating SQL Queries into Relational Algebra, Algorithms for External Sorting, Algorithms for SELECT and JOIN Operations, Algorithms for PROJECT and SET Operations, Implementing Aggregate Operations and Outer Joins.

**Unit V: Disk Storage, File Structures, and Indexing 7 Lecture Hours**

Introduction, Secondary Storage Devices, Buffering of Blocks and Placing File Records on Disk, Operations on Files, Heap Files, Sorted Files, Hashing Techniques, Parallelizing Disk Access using RAID Technology, Secondary Access Paths, Types of Single-Level Ordered Indexes, Multilevel Indexes, Dynamic Multilevel Indexes Using B-Trees and B+ Trees, Indexes on Multiple Keys

**Unit VI: Transaction Management, 8 Lecture Hours**

**Concurrency Control and Recovery**

Introduction to Transaction Processing, Transaction and System Concepts, Desirable Properties of Transactions, Characterizing Schedules based on Recoverability, Characterizing Schedules based on Serializability. Introduction to Concurrency Control, Two Phase Locking Techniques, Concurrency Control on Timestamp Ordering, Validation Concurrency Control Techniques, Granularity of Data items and Multiple Granularity Locking, Recovery Concepts, Recovery Techniques Based on Deferred and Immediate Update, Shadow Paging.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| Textbooks | 1. 1. Ramez Elmasri, and Shamkant B. Navathe, "Fundamentals of Database Systems", 7th Edition, Pearson India, 2017. 2. 2. Raghu Ramakrishnan, "Database Management Systems", 4th Edition, Mcgraw-Hill, 2015. |
| Reference books | 1. 1. A. Silberschatz, H. F. Korth, and S. Sudershan, "Database System Concepts", 6th ed., McGraw Hill, 2013. 2. 2. Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom, "Database Systems-The Complete Book", 2nd Edition, Pearson India Pearson Education, 2011. 3. 3. Pramod J. Sadalage, and Marin Fowler, "NoSQL Distilled: A brief guide to merging world of Polyglot persistence", Addison Wesley, 2012. 4. 4. Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, "MongoDB: The Definitive Guide", 3rd Edition, O'Reilly Media, 2019. |
| Web Resources |  |
| Journals |  |
| MOOCs, online courses |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Database Management Systems Lab**

**List of Experiments**

**Experiment 1:**

**Title:** Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students’ transcripts.

1. The university keeps track of each student’s name, student number, Social Security number, current address and phone number, permanent address and phone number, birth date, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and ZIP Code of the student’s permanent address and to the student’s last name. Both Social Security number and student number have unique values for each student.

a. Each department is described by a name, department code, office number, office phone number, and college. Both name and code have unique values for each department.

b. Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of the course number is unique for each course.

c. Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.

d. A grade report has a student, section, letter grade, and numeric grade (0,1, 2, 3, or 4).

Design an Entity-Relationship diagram for the mail order database and enter the design using a data-modeling tool such as ERWin/free tool. Specify key attributes of each entity type, and structural constraints on each relationship type. Note any unspecified requirements and make appropriate assumptions to make the specification complete.

**Experiment 2**

Title. Consider the following set of requirements for a Company database that is used to keep track of employee.

The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

1. A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
2. We store each employee’s name, Social Security number,2 address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).
3. We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, sex, birth date, and relationship to the employee.

Design an Entity-Relationship diagram for the company database and enter the design

using a data-modeling tool such as ERWin/free tool.

**Experiment 3**

**Title: To understand DDL and DML Command**

**Objective:** To understand the concept of designing issue related to the database with creating, populating the tables. To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key.

1. **Create the tables for** Company database as per ER diagram of Exp 2.

TABLE 1: EMPLOYEE

[ Fname VARCHAR(15) NOT NULL,

Minit CHAR,

Lname VARCHAR(15) NOT NULL,

Ssn CHAR(9) NOT NULL,

Bdate DATE,

Address VARCHAR(30),

Sex CHAR,

Salary DECIMAL(10,2),

Super\_ssn CHAR(9),

Dno INT NOT NULL,

PRIMARY KEY (Ssn),

FOREIGN KEY (Super\_ssn) REFERENCES EMPLOYEE(Ssn),

FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)

]

TABLE 2: DEPARTMENT

[Dname VARCHAR(15) NOT NULL,

Dnumber INT NOT NULL,

Mgr\_ssn CHAR(9) NOT NULL,

Mgr\_start\_date DATE,

PRIMARY KEY (Dnumber),

UNIQUE (Dname),

FOREIGN KEY (Mgr\_ssn) REFERENCES EMPLOYEE(Ssn) );

]

TABLE 3: DEPT\_LOCATIONS

( Dnumber INT NOT NULL,

Dlocation VARCHAR(15) NOT NULL,

PRIMARY KEY (Dnumber, Dlocation),

FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );

TABLE 4: PROJECT

( Pname VARCHAR(15) NOT NULL,

Pnumber INT NOT NULL,

Plocation VARCHAR(15),

Dnum INT NOT NULL,

PRIMARY KEY (Pnumber),

UNIQUE (Pname),

FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );

TABLE 5: WORKS\_ON

( Essn CHAR(9) NOT NULL,

Pno INT NOT NULL,

Hours DECIMAL(3,1) NOT NULL,

PRIMARY KEY (Essn, Pno),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),

FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );

TABLE 6: DEPENDENT

( Essn CHAR(9) NOT NULL,

Dependent\_name VARCHAR(15) NOT NULL,

Sex CHAR,

Bdate DATE,

Relationship VARCHAR(8),

PRIMARY KEY (Essn, Dependent\_name),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );

1. **Insert the following data into their respective tables of Company database.**

| DEPARTMENT | | | |
| --- | --- | --- | --- |
| DNAME | DNUMBER | MGRSSN | MGRSTARTDATE |
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

| EMPLOYEE | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FNAME | LNAME | SSN | BDATE | ADDRESS | SEX | SALARY | SUPERSSN | DNO |
| John | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston TX | M | 30000 | 333445555 | 5 |
| Franklin | Wong | 333445555 | 1965-12-08 | 638 Voss, Houston TX | M | 40000 | 888665555 | 5 |
| Alicia | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring TX | F | 25000 | 987654321 | 4 |
| Jennifer | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire TX | F | 43000 | 888665555 | 4 |
| Ramesh | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble TX | M | 38000 | 333445555 | 5 |
| Joyce | English | 453453453 | 1972-07-31 | 5631 Rice, Houston TX | F | 25000 | 333445555 | 5 |
| Ahmad | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston TX | M | 25000 | 987654321 | 4 |
| James | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston TX | M | 55000 | null | 1 |

| PROJECT | | | |
| --- | --- | --- | --- |
| PNAME | PNUMBER | PLOCATION | DNUM |
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

| WORKS\_ON | | |
| --- | --- | --- |
| ESSN | PNO | HOURS |
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | null |

| DEPENDENT | | | | |
| --- | --- | --- | --- | --- |
| ESSN | DEPENDENT\_NAME | SEX | BDATE | RELATIONSHIP |
| 333445555 | Alice | F | 1986-04-04 | Daughter |
| 333445555 | Theodore | M | 1983-10-25 | Son |
| 333445555 | Joy | F | 1958-05-03 | Spouse |
| 987654321 | Abner | M | 1942-02-28 | Spouse |
| 123456789 | Michael | M | 1988-01-04 | Son |
| 123456789 | Alice | F | 1988-12-30 | Daughter |
| 123456789 | Elizabeth | F | 1967-05-05 | Spouse |

| DEPT\_LOCATIONS | |
| --- | --- |
| DNUMBER | DLOCATION |
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Houston |
| 5 | Sugarland |

**Experiment 4**:

**Title: To understand and apply the concept of Constraints.**

**Objective:** To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key.

**1. Create the tables described below:**

**Table name: CLIENT\_MASTER**

**Description:** used to store client information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Constraints** |
| CLIENTNO | Varchar | 6 | Primary key / first letter must start with ‘C’ |
| NAME | Varchar | 20 | Not Null |
| ADDRESS 1 | Varchar | 30 |  |
| ADDRESS 2 | Varchar | 30 |  |
| CITY | Varchar | 15 |  |
| PINCODE | Integer | 8 |  |
| STATE | Varchar | 15 |  |
| BALDUE | Decimal | 10,2 |  |

**Table Name: PRODUCT\_MASTER Description:** used to store product information

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Attributes** |
| PRODUCTNO | Varchar | 6 | Primary Key/ first letter must start with ‘P’ |
| DESCRIPTION | Varchar | 15 | Not Null |
| PROFITPERCENT | Decimal | 4,2 | Not Null |
| UNIT MEASURE | Varchar | 10 | Not Null |
| QTYONHAND | Integer | 8 | Not Null |
| REORDERL VL | Integer | 8 | Not Null |
| SELLPRICE | Decimal | 8,2 | Not Null |
| COSTPRICE | Decimal | 8,2 | Not Null |

**Table Name: SALESMAN\_MASTER**

**Description:**  used to store salesman information working for the company.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Attributes** |
| SALESMANNO | Varchar | 6 | Primary Key/ first letter must start with ‘S’ |
| SALESMANNAME | Varchar | 20 | Not Null |
| ADDRESS 1 | Varchar | 30 | Not Null |
| ADDRESS 2 | Varchar | 30 |  |
| CITY | Varchar | 20 |  |
| PINCODE | Integer | 8 |  |
| STATE | Varchar | 20 |  |
| SALAMT | Real | 8,2 | Not Null , Cannot be 0 |
| TGTTOGET | Decimal | 6,2 | Not Null , Cannot be 0 |
| YTDSALES | Double | 6,2 | Not Null |
| REMARKS | Varchar | 60 |  |

1. **Insert the following data into their respective tables:**
2. Data for **CLIENT\_MASTER** table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Client no | Name | City | Pincode | State | BalDue |
| C00001 | Ivan bayross | Mumbai | 400054 | Maharashtra | 15000 |
| C00002 | Mamta muzumdar | Madras | 780001 | Tamil nadu | 0 |
| C00003 | Chhaya bankar | Mumbai | 400057 | Maharashtra | 5000 |
| C00004 | Ashwini joshi | Bangalore | 560001 | Karnataka | 0 |
| C00005 | Hansel colaco | Mumbai | 400060 | Maharashtra | 2000 |
| C00006 | Deepak sharma | Mangalore | 560050 | Karnataka | 0 |

1. Data for **PRODUCT**\_**MASTER** table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Product  No | Description | Profit percent | Unit measure | Quantity  On  hand | Recorder  Level | Sell  Price | Cost  Price |
| P00001 | T-Shirt | 5 | Piece | 200 | 50 | 350 | 250 |
| P0345 | Shirts | 6 | Piece | 150 | 50 | 500 | 350 |
| P06734 | Cotton jeans | 5 | Piece | 100 | 20 | 600 | 450 |
| P07865 | Jeans | 5 | Piece | 100 | 20 | 750 | 500 |
| P07868 | Trousers | 2 | Piece | 150 | 50 | 850 | 550 |
| P07885 | Pull Overs | 2.5 | Piece | 80 | 30 | 700 | 450 |
| P07965 | Denim jeans | 4 | Piece | 100 | 40 | 350 | 250 |
| P07975 | Lycra tops | 5 | Piece | 70 | 30 | 300 | 175 |
| P08865 | Skirts | 5 | Piece | 75 | 30 | 450 | 300 |

1. Data for **SALESMAN\_MASTER**  table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Salesman No** | **Name** | **Address1** | **Address2** | **City** | **Pin Code** | **State** |
| S00001 | Aman | A/14 | Worli | Mumbai | 400002 | Maharashtra |
| S00002 | Omkar | 65 | Nariman | Mumbai | 400001 | Maharashtra |
| S00003 | Raj | P-7 | Bandra | Mumbai | 400032 | Maharashtra |
| S00004 | Ashish | A/5 | Juhu | Mumbai | 400044 | Maharashtr(a |

1. **Exercise on retrieving records from a table.**  
   a. Find out the names of all the clients.  
   b. Retrieve the entire contents of the Client\_Master table.  
   c. Retrieve the list of names, city and the state of all the clients.  
   d. List the various products available from the Product\_Master table.  
   e. List all the clients who are located in Mumbai.  
   f. Find the names of salesman who have a salary equal to Rs.3000.
2. **Exercise on updating records in a table**  
   a. Change the city of ClientNo ‘C00005’ to ‘Bangalore’.  
   b. Change the BalDue of ClientNo ‘C00001’ to Rs.1000.  
   c. Change the cost price of ‘Trousers’ to rs.950.00.  
   d. Change the city of the salesman to Pune.
3. **Exercise on deleting records in a table**  
   a. Delete all salesman from the Salesman\_Master whose salaries are equal to Rs.3500.

b. Delete all products from Product\_Master where the quantity on hand is equal to 100.  
c. Delete from Client\_Master where the column state holds the value ‘Tamil Nadu’.

1. **Exercise on altering the table structure**  
   a. Add a column called ‘Telephone’ of data type integer to the Client\_Master table.  
   b. Change the size off SellPrice column in Product \_Master to 10, 2.
2. **Exercise on deleting the table structure along with the data**  
   a. Destroy the table Client\_Master along with its data.

**EXPERIMENT-5**

**Title: To understand and use SQL Sub-Query**

**Objective:** To understand the use of sql subquery.

**1. Create the following table.**

Supplier-(scode,sname,scity,turnover)

Part-(pcode,weigh,color,cost,sellingprice)

Supplier\_Part-(scode,pcode,qty)

**2. Populate the table**

**3. Write appropriate SQL Statement for the following:**

1. Get the supplier number and part number in ascending order of supplier number.

2. Get the details of supplier who operate from Bombay with turnover 50.

3. Get the total number of supplier.

4. Get the part number weighing between 25 and 35.

5. Get the supplier number whose turnover is null.

6. Get the part number that cost 20, 30 or 40 rupees.

7. Get the total quantity of part 2 that is supplied.

8. Get the name of supplier who supply part 2.

9. Get the part number whose cost is greater than the average cost.

10. Get the supplier number and turnover in descending order of turnover.

**EXPERIMENT-6**

**Title: Use of Inbuilt functions and relational algebra operations**

**Objective:** To understand the use of built-in functions and relational algebra with SQL queries.

Write and execute the following queries using the Relational Algebra on the COMPANY database schema.

1. Retrieve the names of all employees in department 5 who work more than 10 hours
2. per week on the ‘ProductX’ project.
3. List the names of all employees who have a dependent with the same first name as
4. themselves.
5. Find the names of employees who are directly supervised by ‘Franklin Wong’.
6. Retrieve the names of employees who work on every project.
7. Retrieve the names of employees who do not work on any project.
8. Retrieve the names and addresses of all employees who work on at least one project
9. located in Houston but whose department has no location in Houston.
10. Retrieve the last names of all department managers who have no dependents.

**EXPERIMENT-7**

**Title: Use of Inbuilt functions and relational algebra operations**

**Objective:** To understand the use of built-in functions and relational algebra with SQL queries.

1. **Create the following two tables (EMP and DEPT)**

**EMP TABLE**

|  |
| --- |
| EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO |
| 7369 SMITH CLERK 7902 17-DEC-80 500 800 20 |
| 7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30 |
| 7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30 |
| 7566 JONES MANAGER 7839 02-APR-81 2975 20 |
| 7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30 |
| 7698 BLAKE MANAGER 7839 01-MAY-81 2850 30 |
| 7782 CLARK MANAGER 7839 09-JUN-81 2450 10 |
| 7788 SCOTT ANALYST 7566 09-DEC-82 3000 20 |
| 7839 KING PRESIDENT 17-NOV-81 5000 10 |
| 7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30 |
| 7876 ADAMS CLERK 7788 12-JAN-83 1100 20 |
| 7900 JAMES CLERK 7698 03-DEC-81 950 30 |
| 7902 FORD ANALYST 7566 03-DEC-81 3000 20 |
| 7934 MILLER CLERK 7782 23-JAN-82 1300 10 |

**DEPT TABLE**

DEPTNO DNAME LOC

---------- -------------- -------------

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

Write the Queries for the following using built-in functions.

1. Retrieve the average salary of all employees.
2. Retrieve the number of employees.
3. Retrieve a distinct number of employees.
4. Retrieve the total salary of the employee group by job.
5. Display the employee information with the maximum salary.
6. Find the highest-paid employee in department 10.
7. List the employees whose salary is equal to the average of the maximum and minimum.
8. List the employees who joined in the company on the same date.
9. Display the employee names in upper and lower case.
10. Find the date 3 days later from hiredate.

**EXPERIMENT-8**

**Title: Use of different SQL clauses and joins**

**Objective:** To understand the use of GROUP BY and the HAVING clause and execute the SQL commands using JOIN

1.Consider the following schema:

**Student (sid, sname, age)**

**Match (mid, mname, venue)**

**Play (sid, mid, day(date))**

1. Populate all the tables.
2. nFind all information of students who have played match number B10.
3. Find the name of matches played by Amit.
4. Find the names of students who have played a match in Delhi.
5. Find the names of students who have played at least one match.
6. Find the ids and names of students who have played two different matches on the same day.
7. Find the ids of students who have played a match in Delhi or Mumbai.
8. Find the average age of students.

**EXPERIMENT-9**

**Title: To understand the concepts of Views.**

**Objective:** Students will be able to implement the concept of views.

**1. Create table of table name: EMPLOYEES and add 6 rows**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Width | Attributes |
| Employee\_id | Character | 10 | PK |
| First\_Name | Character | 30 | NN |
| Last\_Name | Character | 30 | NN |
| DOB | Date |  |  |
| Salary | Number | 25 | NN |
| Department\_id | Character | 10 |  |

**2. Execute the following view related queries:**

1. Create View of name emp\_view and the column would be Employee\_id, Last\_Name, salary and department\_id only.
2. Insert values into view(remove the NOT NULL constraint and then insert values):
3. Modify, delete and drop operations are performed on view.
4. Creates a view  named salary\_view. The view shows the employees in department 20 and their annual salary.

**EXPERIMENT-10**

**Title: Create the following views in SQL on the COMPANY database schema presented in Experiment 2.**

1. A view that has the department name, manager name, and manager salary for every department.
2. A view that has the employee name, supervisor name, and employee salary for each employee who works in the ‘Research’ department.
3. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.
4. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

**EXPERIMENT-11**

**Title: To understand the concepts of Index.**

**Objective:** Students will be able to implement the concept of index.

**Create table of table name: EMPLOYEES and add 6 rows**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Width | Attributes |
| Employee\_id | Character | 10 | PK |
| First\_Name | Character | 30 | NN |
| Last\_Name | Character | 30 | NN |
| DOB | Date |  |  |
| Salary | Number | 25 | NN |
| Department\_id | Character | 10 |  |

**1. Execute the following index related queries:**

1. Create an index of name employee\_idx on EMPLOYEES with column Last\_Name, Department\_id
2. Find the ROWID for the above table and create a unique index on employee\_id column of the EMPLOYEES.
3. Create a reverse index on employee\_id column of the EMPLOYEES.
4. Create a unique and composite index on employee\_id and check whether there is duplicity of tuples or not.
5. Create Function-based indexes defined on the SQL functions UPPER(column\_name) or LOWER(column\_name) to facilitate case-insensitive searches(on column Last\_Name).
6. Drop the function based index on column Last\_Name.

**EXPERIMENT-12**

**Title: To understand the concepts of Sequence.**

**Objective:** Students will be able to implement the concept of sequence.

1. Create a sequence by name EMPID\_SEQ starting with value 100 with an interval of 1.
2. Write a SQL command for finding the current and the next status of EMPID\_SEQ.
3. Change the Cache value of the sequence EMPID\_SEQ to 20 and maxvalue to 1000.
4. Insert values in employees table using sequences for employee\_id column.
5. Drop sequence EMPID\_SEQ.
6. Create a sequence called REVERSE to generate numbers in the descending order from 10000 to 1000 with a decrement of 5.

**EXPERIMENT-13**

**Title: To understand the concepts of PL/SQL programming.**

**Objective:** Students will be able to implement the basic concepts of Pl/SQL.

1. Write a PL/SQL code to accept the value of A, B & C display which is greater.
2. Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times.
3. Write a PL/SQL code block to find the factorial of a number.
4. Write a PL/SQL program to generate Fibonacci series.
5. Write a PL/SQL code to fund the sum of first N numbers

**EXPERIMENT-14**

**Title: To understand the concepts of function and procedure in PL/SQL.**

**Objective:** Students will be able to implement the Pl/SQL programs using function and procedure.

1. Implement the above experiments of PL/SQL using functions and procedures.

**EXPERIMENT-15**

**Title: To understand the concepts of implicit and explicit cursor.**

**Objective:** Students will be able to implement the concept of implicit and explicit cursor.

1. Using implicit cursor update the salary by an increase of 10% for all the records in EMPLOYEES table, and finally display how many records have been updated. If no records exist display the message “**No Change**”.
2. Using explicit cursor fetch the employee name, employee\_id and salary of all the records from EMPLOYEES table.
3. Using explicit cursor Insert the records from EMPLOYEES table for the columns employee\_id, Last\_Name and salary for those records whose salary exceeds 2500 into a new table TEMP\_EMP

**EXPERIMENT-16**

**Title: To understand the concepts of Trigger.**

**Objective:** Students will be able to implement the concept of trigger.

**CUSTOMER Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | NAME | AGE | ADDRESS | SALARY |
| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |
| 2 | Khilan | 25 | Delhi | 1500.00 |
| 3 | Kaushik | 23 | Kota | 2000.00 |
| 4 | Chaitali | 25 | Mumbai | 6500.00 |
| 5 | Hardik | 27 | Bhopal | 8500.00 |
| 6 | Komal | 22 | MP | 4500.00 |

1. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values.

**EXPERIMENT-17**

**Title: To understand the concepts of Trigger.**

**Objective:** Students will be able to implement the concept of trigger.

1. CREATE TRIGGER SALARY\_VIOLATION BEFORE INSERT OR UPDATE OF SALARY, SUPERVISOR\_SSN ON EMPLOYEE of experiment 3

**Total Lab hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Ivan Bayross, "SQL, PL/SQL – The Programming Language of Oracle", 4th Revised Edition, 2010. 2. Kristina Chodorow, and Michael Dirolf, "MongoDB: The Definitive Guide", O'Reilly Media Inc., 2010. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG2006** | **Discrete Mathematical Structures** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 7** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

To develop mathematical reasoning skills that equip the students with ideas and techniques, required in Discrete Mathematics and its Applications, necessary for understanding and practicing the art and science of computing.

**Course Outcomes**

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

**CO1**. Develop a comprehensive understanding of discrete structures, viz., mathematical logic, propositions, connectives, arguments, sets, functions, relations, etc., that build the foundation of computer science.

**CO2**. Recognize the applicability of the basic notions of number theory and counting principles such as modular arithmetic, congruence equations, Pigeonhole principle, principle of inclusion and exclusion, etc., in solving complex problems in diverse domains related to cryptography and internet security.

**CO3**. Illustrate the basic terminologies in graph theory which are used in design of algorithms for a wide range of applications related to shortest path problems, graph coloring problems, etc.

**CO4.** Use the framework of network science in building models for various problems arising in communication, social and biological networks.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 3 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 2** | 3 | 3 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 3** | 3 | 2 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **CO 4** | 3 | 2 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |
| **Average** | 3 | 2.5 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Motivation 1 Lecture Hour**

Introduction to Discrete Mathematics and its significance in Computer Science. Why learn Discrete Mathematics?

**Unit II: Mathematical Logic** **6 Lecture Hours**

Mathematical Logic, Proposition, Connectives, Tautologies, and Contradictions, Logical Equivalences, Normal forms, and applications. Predicates and Quantifiers, Interpretation of an Argument, and its validity. Recapitulation of the Unit.

**Unit III: Relations**  **6 Lecture Hours**

Review of Set Theory, Functions, one-one onto Functions, Relations and their properties, n-ary Relations, Illustrations through Applications, Representations, Closures, Partial Ordering, Decomposition Theorems for Partial Orders, Posets and Lattices. Recapitulation of the Unit.

**Unit IV: Number Theory** **5 Lecture Hours**

Modular Arithmetic, Primes, Fundamental Theorem of Arithmetic, GCD/LCM, Euclidean Algorithm, Bézout’s Identity, Solving Congruences, Linear Diophantine Equation, Chinese Remainder Theorem, Fermat’s Little Theorem, Illustration through Examples in Cryptography and Internet Security, Discrete Log, Orthogonal Latin Squares. Recapitulation of the Unit

**Unit V: Induction and Basic Counting Principles 10 Lecture Hours**

Induction, Strong Induction, Well-ordering Principle, Recursive Definitions and Structural Induction, Pigeonhole Principle, Binomial and Multinomial Coefficients and Identities, Elementary Applications to Discrete Probability, Recurrence Relations and Equations, Generating Function Techniques, Principles of Inclusion and Exclusion and their Applications. Recapitulation of the Unit.

**Unit VI: Graph Theory 10 Lecture Hours**

Graphs and Graph Models, Basic Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Vertex and Edge Connectivity, Adjacency and Incidence Matrices, Konigsberg Bridge Problem, Euler Graph, Travelling Salesman Problem, Hamiltonian Graphs, Shortest-Path Problems, Planar Graph, Euler’s Formula, Graph Coloring, Welch Powell Algorithm, Decomposition Algorithm, Random Graphs. Recapitulation of the Unit.

**Unit VII: Network Science 7 Lecture Hours**

Illustrative Examples of Real Networks, Network and Graph, Degree Distribution, Power-Law and Scale Free Property, Sparsity in Real Networks, Paths and Distances, Network Diameter, Connectedness and Clustering Coefficient, Random Network, Evolution of a Random Network, Small World Network: Watts-Strogatz Model, Emergence of Network Science. Recapitulation of the Unit.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Rosen, K. H., "Discrete Mathematics and its Applications", 7th Edition, McGraw Hill, 2017. (Units - II, III, IV, VI) 2. R. C. Bose, and B. Manvel, "Introduction to Combinatorial Theory", Wiley Series in Probability and Mathematical Statistics, 1984. (Units - V) 3. Albert-László Barabási, "Network Science", 1st Edition, Cambridge University Press, 2016. (Unit-VI, VII) |
| **Reference books** | 1. W. K. Grassmann, and J-P Tremblay, "Logic and Discrete Mathematics: A Computer Science Perspective", 1st Edition, Pearson, 1995. 2. E. Lehman, F. T. Leighton, and A. R. Meyer, "Mathematics for Computer Science", Samurai Media Limited, 2017. 3. D. B. West, "Introduction to Graph Theory", 2nd Edition, Pearson Singapore, 2000. 4. J. L. Hein, "Discrete Structure, Logic, and Computability", 3rd Edition, Jones and Barlett Publishers, 2010. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **MATH2059** | **Linear Algebra** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The course aims to

1. Provide students with understanding of fundamental concepts of linear algebra and their applications.

2. Develop mathematical models employing Linear Algebra framework for problems arising in a variety of disciplines.

3. Empower the students to learn and formulate problems using linear Algebra in science, engineering including emerging areas like data analytics and deep learning.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Model situations in diverse contexts involving vectors, matrices, and systems of linear equations.

**CO2.** Demonstrate an interplay between the core mathematical concepts and applications in computer and allied sciences.

**CO3.** Comprehend and visualize concepts of eigenvalues and eigenvectors in computer graphics and emerging applications.

**CO4.** Apply SVD, a powerful technique, which is crucial for various computational tasks, including machine learning, computer graphics, data compression, and image processing.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **2** | **2** | **3** | **-** | **-** | **-** | **-** | **2** | **-** | **-** | **3** | **-** | **-** | **-** |
| **CO 2** | **1** | **2** | **3** | **2** | **-** | **-** | **-** | **-** |  | **-** | **-** | **3** | **2** | **-** | **-** |
| **CO 3** | **2** | **3** | **3** | **3** | **2** | **2** | **-** | **-** | **2** | **-** | **-** | **3** | **3** | **2** | **-** |
| **CO 4** | **3** | **3** | **3** | **3** | **2** | **-** | **-** | **2** | **2** | **-** | **2** | **3** | **2** | **2** | **-** |
| **Average** | **2** | **2.5** | **2.75** | **2.75** | **1** | **.5** | **-** | **.5** | **1.5** | **-** | **.5** | **3** | **1.75** | **1** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Linear Algebra 1 Lecture Hours**

Linearity vs Non-linearity (Real life examples), Panoramic view of linear algebra(Linear + algebra), Mention of emerging Applications of Computer Science: Google PageRank, Google Maps, etc.

**Unit II: System of Linear Equations 8 Lecture Hours**

Vectors and linear combination, Visualization of the system of linear equations, Elimination using matrices, Rank of a matrix, Echelon forms, Normal form, Solution of a homogeneous and non-homogeneous system of equations, Applications in emerging areas: Machine learning models, Cryptography and Color models, Recap of Unit-II.

**Unit III: Vector Spaces and Linear Transformation 12 Lecture Hours**

Vector spaces and subspaces, Linear span, Linear independence/dependence of vectors, Basis and dimension, Linear mapping, Matrix as a linear mapping, Kernel and image of linear mapping, Null space, Rank nullity theorem, Singular and non-singular mappings, Isomorphisms, Operations with linear mappings, Similarity of matrices, Change of basis, Inner product spaces, Vector and matrix norms, Orthogonality, Orthogonal sets and bases, Projections, Gram-Schmidt orthogonalization process, Applications of linear transformation: Data Smoothing, Image scaling, Recap of Unit III.

**Unit IV: Eigenvalues and Eigenvectors 8 Lecture Hours**

Polynomial of matrices, Characteristic polynomial, Cayley-Hamilton theorem, Eigenvalues and eigenvectors, Geometric interpretation of eigenvectors, Diagonalization, Power of a matrix, Function of matrices, Diagonalization of symmetric matrices, Quadratic forms, Methods for computing Eigenvalues, Method of Least squares, Application of Eigensystems: Facial and Ear recognition, Feature extraction, Internet search engines. Recap of Unit IV.

**Unit V: Singular value decomposition 8 Lecture Hours**

Spectral decomposition, Singular value decomposition (SVD), Best rank k approximations, Power method for computing the Singular value decomposition, Applications of Singular value decomposition: Principal component analysis, Singular vectors, Centering data, Ranking documents and Web pages, Clustering, Recap of Unit V.

**Unit VI: Applications of Linear Algebra: 8 Lecture Hours**

**Computational Approach**

Applications of linear systems: Design of traffic patterns/Circuit with one closed loop/Balancing chemical equations; Data compression using SVD, Word Embeddings and Exploring Biases in Data, Markov Matrices and Applications to PageRank, Game of strategy, Vector space models for information retrievals, Vector matrix: Moves on a chessboard, Distribution of genotypes in a population.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. G. Strang, "Linear Algebra and its Applications", 4th Edition, Cengage Learning, 2005. 2. G. Williams, "Linear Algebra with Applications", 8th Edition, Jones and Bartlett Learnings, 2012. 3. H. Anton, and C. Rorres, "Elementary linear algebra with supplemental applications", 11th Edition, Wiley, 2016. |
| **Reference books** | 1. I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning", The MIT Press, 2016. 2. K. Singh, "Linear Algebra Step by Step", Oxford University Press, 2013. 3. D. C. Lay, Steven R. Lay, and Judi J. McDonald, "Linear Algebra and its Applications", 5th Edition, Pearson Education India, 2023. 4. H. Wendland, "Numerical Linear Algebra An Introduction", Cambridge University Press, 2018. 5. W. Ford, "Numerical linear algebra with applications using MATLAB", Academic Press, 2014. [Chapter 7, 15, 17] 6. J. MacCormick, "Nine Algorithms that Changed the Future", Princeton University Press, 2021. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SLLS2004** | **Indian Constitution** | | 0 | 0 | 0 | 0 |
| **Total Units to be Covered: 04** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The course aims to

1. To realise the significance of constitution of India to students from all walks of life and help them to understand the basic concepts of Indian constitution.
2. To identify the importance of fundamental rights as well as fundamental duties.
3. To understand the functioning of Union, State and Local Governments in Indian federal system.
4. To learn procedure and effects of emergency, composition and activities of election commission and amendment procedure.

**Course Outcomes**

**CO1.** Understand and explain the significance of Indian Constitution as the fundamental law of the land.

**CO2.** Exercise his fundamental rights in proper sense at the same time identifies his responsibilities in national building.

**CO3**. Analyse the Indian political system, the powers and functions of the Union, State and Local Governments in detail

**CO4.** Understand Electoral Process, Emergency provisions and Amendment procedure.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 4** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **Average** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Constitution:**

Meaning and importance of the Constitution, salient features of Indian Constitution. Preamble of the Constitution**.** Fundamental rights**-** meaning and limitations. Directive principles of state policy and Fundamental duties -their enforcement and their relevance.

**Unit II: Union Government:**

Union Executive- President, Vice-president, Prime Minister, Council of Ministers. Union Legislature- Parliament and Parliamentary proceedings. Union Judiciary-Supreme Court of India – composition and powers and functions.

**Unit III: State and Local Governments:**

State Executive- Governor, Chief Minister, Council of Ministers. State Legislature-State Legislative Assembly and State Legislative Council. State Judiciary-High court. Local Government-Panchayat raj system with special reference to 73rd and Urban Local Self Govt. with special reference to74th Amendment.

**Unit IV: Election provisions, Emergency provisions, Amendment of the constitution**

Election Commission of India-composition, powers and functions and electoral process. Types of emergency-grounds, procedure, duration and effects. Amendment of the constitution- meaning, procedure and limitations.

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1 Ethics and Politics of the Indian Constitution Rajeev Bhargava Oxford University Press, New Delhi, 2008 2. The Constitution of India B.L. Fadia Sahitya Bhawan; New edition (2017) 3. Introduction to the Constitution of India DD Basu Lexis Nexis; Twenty-Third 2018 edition |
| **Reference books** | 1**.** Introduction to Indian Knowledge System: Concepts and Applications, authored by Dr. B. Mahadevan, Professor of IIM Bangalore and founding Vice Chancellor of Chinmaya Vishwa Vidyapeeth; Dr. Vinayak Rajat Bhat, Associate Professor, Chanakya University, Bengaluru; and Dr. Nagendra Pavana R.N., Faculty at the School of Vedic Knowledge Systems, Chinmaya Vishwa Vidyapeeth; Forewords by Dr. Anil Sahasrabudhe, Former Chairman AICTE; Prof. Subhash Kak, Oklahoma State University, USA; and Dr. S. Sadagopan, Chairman, BoG, IIITDM – Kancheepuram & Founder Director, IIITBangalore |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme** continuous assement **/**NPTEL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **EMPL002** | **EDGE-SoftSkills** | | 1 | 0 | 0 | 0 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Employment training related course.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Exploratory-2** | | 3 | 0 | 0 | 3 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SFLS0004** | **Communication Skills (Written)** | |  |  |  | 2 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**SEMESTER V**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG3040** | **Cryptography and Network Security** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** | Data Communication and Networks - CSEG2065 | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the fundamental concepts of cryptography and its applications.
2. Learn about various encryption and decryption techniques.
3. Explore network security protocols and mechanisms.
4. Develop the skills to analyze, design, and implement secure communication systems.

**Course Outcomes**

On completion of this course, the students will be able to

1. Relate the historical development of cryptography, its basic concepts, and terminology, and identify the different types of attacks and security services in a networked environment.
2. Illustrate the principles of symmetric-key cryptography and Public Key Cryptography, including the working of algorithms like DES, AES, and RSA and evaluate the security of block cipher modes of operation and its applications.
3. Use cryptographic hash functions, such as MD5, SHA-1, and SHA-256, for ensuring data integrity, and design applications that implement HMAC for message authentication.
4. Demonstrate an understanding of network security principles, including authentication, access control, and various network attacks, and propose countermeasures to mitigate those attacks.
5. Design and implement secure communication systems using cryptographic techniques and network security principles, and critically analyze the effectiveness of these systems in real-world scenarios.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 2** | 2 | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 3** | 2 | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 4** | 2 | 2 | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 5** | 2 | 2 | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **Average** | 2 | 2 | 1.8 | 0.8 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Unit I: Introduction to Cryptography 5 Lecture Hours**

Historical development of cryptography, Basic concepts and terminology, Types of attacks and security services, Classical cryptography and its limitations, stream and block ciphers, cryptanalysis, stereography, prime and relative prime numbers, modular arithmetic, Fermat’s and Euler’s theorem, primality testing, Euclid’s Algorithm.

**Unit II: Symmetric-Key Cryptography 10 Lecture Hours**

Substitution and transposition ciphers, Shannon’s theory of confusion and diffusion, fiestal structure, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), Block cipher modes of operation, traffic confidentiality, key distribution, random number generation.

**Unit III: Introduction to Number Theory and Public-Key Cryptography**

**10 Lecture Hours**

Introduction to graph, ring and field, Introduction to public-key cryptography, RSA algorithm, Diffie-Hellman key exchange, introductory idea of Elliptic curve cryptography, Digital signatures, and certificates

**Unit IV: Cryptographic Hash Functions 10 Lecture Hours**

Properties and applications of hash functions, Message Digest algorithms (MD5, SHA-1, SHA-256), HMAC (Hash-based Message Authentication Code), Kerberos and X.509, directory authentication service.

**Unit V: Network Security** **10 Lecture Hours**

Secure communication principles, Authentication and access control, Network attacks and countermeasures, Virtual Private Networks (VPNs), Transport Layer Security (TLS) and Secure Sockets Layer (SSL), Wireless Network Security, Security in Mobile and Cloud Computing

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. William Stallings, "Cryptography and Network Security: Principles and Practice", 7th Edition, Pearson, 2017. 2. Christof Paar, and Jan Pelzl, "Understanding Cryptography: A Textbook for Students and Practitioners", Springer, 2014. |
| **Reference books** | 1. Wade Trappe, and Lawrence C. Washington, "Introduction to Cryptography with Coding Theory", 2nd Edition, Pearson, 2005. 2. William Stallings, "Network Security Essentials: Applications and Standards", 6th Edition, Pearson, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG3055** | **Formal Languages and Automata Theory** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 4** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** | **Discrete Mathematical Structures - CSEG2006** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the foundations and principles of formal languages used in computing.
2. Formulate formal and rigorous mathematical arguments.
3. Prove or disprove theorems in automata theory.
4. Develop a formal computational model pertaining to real-world problems.

**Course Outcomes**

On completion of this course, the students will be able to,

1. Recognize the various language classes and their relationships.
2. Comprehend regular grammar and expressions using finite automata.
3. Develop grammar and recognizer for various formal languages.
4. Analyze the decidability and intractability of different computational problems.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | **-** | **-** |
| **CO 2** | 3 | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | **-** |
| **CO 3** | 2 | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | **-** |
| **CO 4** | 2 | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | **-** | **-** |
| **Average** | 2.25 | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2.5 | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Finite Automata & Regular Languages 15 Lecture Hours**

Computational problems vs. Formal Languages. Finite State Automata- Deterministic and Non-Deterministic, Regular Expression and Language, Closure property of Regular Language, Limitations of Regular Language, Pumping Lemma, Minimization Algorithm, Myhill-Nerode relations, Finite Automata with outputs- Mealy and Moore Machines.

**Unit II: Grammar’s Classification and Push Down Automata 15 Lecture Hours**

Grammars and Chomsky Classification, Regular Grammar, Context Free Grammar and Languages: Ambiguity, Simplification of CFGs, Normal Forms for CFGs, Pumping Lemma for CFLs, Applications to Parsing, Pushdown Automata (PDA), PDA vs CFLs, Deterministic CFLs, Linearly Bounded Automata (LBA).

**Unit III: Turing Machine 10 Lecture Hours**

Introduction to Turing Machines, Configurations, Multi-tape Turing machines, Halting vs Looping, Recursive and Recursively Enumerable Languages, Decidable and Undecidable Languages.

**Unit IV: Decidability & Intractability 5 Lecture Hours**

Undecidability of Halting Problem, Introduction to the Theory of NP-completeness, Reductions, Rice theorem, Post Correspondence Problem, Church-Turing Thesis, Cook-Levin Theorem.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. John E. Hopcroft, Rajiv Motwani, and Jeffrey D. Ullman, "Automata Theory, Languages, and Computation", 3rd Edition, Pearson, 2008. 2.Dexter C. Kozen, "Automata and Computability", Springer Publishers, 2007. |
| **Reference books** | 1. Peter Linz, "An Introduction to Formal Languages and Automata", 6th edition, Jones & Bartlett India, 2016. 2. Manish Kr. Jha, "Automata Theory", 2nd Edition, S.Chand Publication, 2015. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG3002** | **Object Oriented Analysis and Design** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** | **Software Engineering - CSEG2064** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Learn object-oriented analysis and design and UML diagrams.
2. Apply the UML to solve several common modelling problems.
3. Learn to apply object-oriented concepts to all stages of the software development.

**Course Outcomes**

On completion of this course, the students will be able to

1. Describe OOAD Principles including encapsulation, inheritance, and polymorphism, and comprehend their relevance in software development.
2. Illustrate UML Modeling and the role of actors, classes, and objects in a software design
3. Illustrate activity, interaction, event, signals to design state diagram.
4. Modeling software systems using OOAD methodologies to tackle complex real-world challenges.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **3** | **2** | **3** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **-** |
| **CO 2** | **3** | **2** | **3** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **-** |
| **CO 3** | **3** | **3** | **2** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **-** |
| **CO 4** | **3** | **2** | **2** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **-** |
| **Average** | **3** | **2.25** | **2.5** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: OOAD Basics 10 Lecture Hours**

Overview of object oriented system development, Basic notion of objects, Object-oriented concepts: objects, classes, encapsulation, inheritance, polymorphism, Benefits of object-oriented analysis and design, Object basics –The unified process, Multiple Views of Objects, Modelling concepts, Modeling as a design technique, Analysis and modelling, UML and its role in software development, UML diagrams - Class diagrams, use case diagrams, Class, State, Interaction Modelling sequence diagrams, Modeling relationships between classes and objects, UML Rational Unified Process(RUP) Contrasting with Procedural Computation, Client-Server/Message Passing.

**Unit II: Basic & Advanced Structural Modeling 10 Lecture Hours**

Requirement Modeling: Requirement Engineering, Requirement Modeling: Use Cases Diagrams, Basic structural Modeling: Class diagrams, their components, Modeling Relationships, Common Mechanisms, Diagrams, abstract classes, Interfaces, Types and Roles, Packages. Class & object Diagrams: Terms, concepts, Modeling techniques for class & object diagrams Identifying operations and Specifying operations using CRC card.

Advanced Structural Modeling Concepts: design patterns, architectural modeling, and framework integration, Exploring advanced UML concepts like constraints, derived properties, and visibility control. Overview of C++: Procedural Extension of C, Objects, Classes and Encapsulation, Overloading, Inheritance & Polymorphism, Type Casting.

**Unit III: Basic Behavioral Modeling 10 Lecture Hours**

Basic Behavioral Modeling-I: Use Case Modeling, Use Case Diagrams using UML notation, use case diagram extensions and variations, Use Case Descriptions, Activity Diagrams, State Diagrams, Swim lanes Design Architecture

Basic Behavioral Modeling-II: Interaction, Interaction Diagrams, Behavioral Patterns, Scenario-Based Analysis, Verification and Validation of Behavior Models

**Unit IV: Advanced Behavioral Modeling 10 Lecture Hours**

Advanced Behavioral Modeling: Advanced Use Case Modeling , Activity Diagrams, State Machine Diagrams, Interaction Diagrams: Sequence and Communication Diagrams, Event-Driven Architectures, State Machines, Processes and Threads, Time and Space, State chart diagrams & Sequential Diagrams, Business Process Modeling, Modeling Reactive Systems, Advanced Behavioral Patterns, Case Studies and Practical Applications: Analyzing and modeling real-world systems using advanced behavioral modeling techniques, Case studies on event-driven systems, business processes, and reactive systems, Integration of advanced behavioral models with software development practices

**Unit V: Advanced Architectural Modeling 5 Lecture Hours**

Component Diagrams, Deployment Diagrams, Architectural Styles (client-server, microservices, event-driven), Architectural patterns (e.g., MVC, MVVM, layered architecture), Architectural Decision-Making: Enterprise Architecture, Distributed Systems Architecture - Case studies of distributed system architectures (e.g., cloud computing, IoT platforms), Domain-Driven Design (DDD), Service-Oriented Architecture (SOA, Event-Driven Architecture (EDA), Cloud and Serverless Architectures, Security and Privacy in Architectural Modeling, Emerging Trends and Technologies in Architecture: containerization, microservices, and serverless computing, Case studies and discussions on the practical application of new architectural approaches

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Michael Blaha and James Rumbaugh, “Object-oriented modeling and design with UML”, 2nd Edition, Prentice-Hall of India, 2007. 2. Grady Booch, James Rumbaugh, and Ivar Jacobson, "The Unified Modeling Language User Guide", 2nd Edition, Pearson Education, 2012. 3. Russ Miles, and Kim Hamilton, "Learning UML 2.0: A Pragmatic Introduction to UML", O'Reilly, 2006. |
| **Reference books** | 1. Meilir Page-Jones, "Fundamentals of Object-Oriented Design in UML", Pearson Education, 2002. 2. Pascal Roques, "Modeling Software Systems Using UML2", WILEY- Dreamtech India Pvt. Ltd, 2004. 3. Atul Kahate, "Object Oriented Analysis & Design", The McGraw-Hill Companies, 2004. 4. John W. Satzinger, Robert B. Jackson, and Stephen D Burd, "Object-Oriented Analysis and Design with the Unified Process", Cengage Learning, 2007. 5. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, "UML 2 Toolkit", WILEY-Dreamtech India Pvt. Ltd, 2003. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG3056** | **Probability, Entropy, and MC Simulation** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

This course envisions to impart into the students to:

1. Develop a solid foundation in probability, enabling them to recognize probability distributions in real-world scenarios and effectively address probabilistic problems.
2. Equipping with the necessary skills to proficiently utilize Monte Carlo algorithms in tackling intricate problems that involve uncertainty and randomness.
3. Enhances students' analytical and computational abilities to tackle practical applications within the realm of computer science.
4. Possess the requisite knowledge and tools to confidently approach real-world problems that demand expertise in probabilistic analysis, information processing, and simulation techniques.

**Course Outcomes**

On completion of this course, the students will be able to

1. Identify appropriate probability distributions, discrete / continuous, in problems with uncertainty and compute quantities of interest.
2. Analyze multivariate data and draw statistical information by identifying the appropriate random variate.
3. Gain understanding of probabilistic behavior of complex uncertain systems through Monte-Carlo simulation experiments.
4. Calculate uncertainty in a random experiment using entropy function and learn to compute the distance between two probability distributions using KL divergence.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 1 | **-** |
| **CO 2** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 1 | **-** |
| **CO 3** | 2 | 2 | 1 | 2 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 1 | **-** |
| **CO 4** | 2 | 2 | 1 | 2 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 1 | **-** |
| **Average** | 2 | 2 | 1 | 1 | .5 | .5 | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 1 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit 0: Motivation 1 Lecture Hours**

Randomness/Uncertainty all pervasive, Probability theory gives rational description while entropy provides quantitative measure of uncertainty, Generation of random numbers for conducting numerical experiments for simulations.

**Unit I: Probability 8 Lecture Hours**

Motivation: Frequency based probability, Probability models, Sample space, Algebra of events, Notion of statistical regularity, Simpson’s Paradox, Axioms of probability, Some probability propositions, Conditional probability, Independence of events, Birthday problem, How to make a fair coin from a biased coin?, Reliability: Series and parallel systems, Theorem of total probability, Bayes’ theorem, Debate: Frequentist vs. Bayesian, Bernoulli trials, Geometric probability – continuous sample space. Recap of Unit I.

**Unit II: Discrete Random Variables 8 Lecture Hours**

Definition of random variables, Distribution function, Discrete RV, Probability mass function, Expectation, Mean and variance, Moment generating function. Special discrete distributions: Bernoulli and Binomial, Poisson, Geometric and memoryless property of Geometric pmf, Negative binomial, Hypergeometric, Uniform. Indicator random variable. Recursive formulation for binomial probabilities. Recap of Unit II.

**Unit III: Continuous Random Variables 9 Lecture Hours**

Definition, Distribution function, Probability density function, Expectation, Mean and variance, Moment generating function. Special continuous distributions: Uniform, Exponential, Gamma, Normal, Pareto. Weibull distribution, Reliability, and failure rate. Functions of random variable. Markov inequality and Chebyshev’s inequality. Limiting distributions and Stirling’s approximation. Recap of Unit III.

**Unit IV: Jointly Distributed Random Variables 7 Lecture Hours**

Random vectors, Joint probability distribution, Independent random variables, Sum of independent random variable, Conditional distribution, Conditional expectation, Expected number of comparisons in Quick sort, Covariance, Correlation coefficient, Multivariate normal distribution. Limit theorems: Law of large numbers and Central limit theorem, Breakdown of CLT. Recap of Unit IV.

**Unit V: Monte-Carlo (MC) Simulation 7 Lecture Hours**

MC methods – sampling and simulation, Analogy between probability and volume, Estimation of , Buffon’s needle problem, Estimation of integrals, Linear congruential methods, Pseudo random numbers, Random variable generation - Inverse transform method, Acceptance rejection method. Generating continuous random variables: Exponential, Normal - Box-Muller approach, Gamma distribution. Generating discrete random variables: Binomial and Poisson variables. Illustration of CLT using random numbers. MC simulation of some probability models. Recap of Unit V.

**Unit VI: Information Theory and its Applications 5 Lecture Hours**

Overview of information theory, Information / Surprise and Entropy, Entropy as a measure of randomness, Properties of Entropy function, Applications of Entropy: English language, music, logical problems, lossless compression. Kullback-Leibler measure of divergence. Recap of Unit VI.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. S. M. Ross, "A first course in probability", Pearson, 2020. 2. K. S. Trivedi, "Probability & statistics with reliability, queuing and computer science applications", John Wiley & Sons, 2008. Chapters 1 to 5. 3. R. Y. Rubinstein, and D. P. Kroese, "Simulation and the Monte Carlo method", John Wiley & Sons, 2016. |
| **Reference books** | 1. M. Mitzenmacher, and E. Upfal, "Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis", Cambridge University Press, 2017. 2. R. Nelson, "Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling", Springer Science & Business Media, 2013. 3. I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning", MIT press, 2016. Chapter 3. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Exploratory-3** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SFLS0005** | **Working with People** | | **2** | **0** | **0** | **2** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **EMPL003** | **EDGE – Advance Communication** | |  |  |  | **0** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Employment training related course

**SEMESTER VI**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG3057** | **Statistics and Data Analysis** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG2064** | **Software Engineering** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To explore software development methodologies (waterfall, agile, DevOps) and their integration of testing, quality assurance, reliability, and risk management.
2. To comprehend software requirements engineering and develop skills in creating well-structured Software Requirements Specifications (SRS).
3. To acquire understanding of planning a software project, its cost estimation models and to understand the software quality models.

**Course Outcomes**

1. Understand the fundamental concepts and importance of Software Engineering in modern software development.
2. Learn various software development methodologies, including Agile, Waterfall, and iterative approaches.
3. Explore software design principles and architectural patterns for creating robust and maintainable software systems.
4. Apply project management principles to effectively plan, monitor, and control software projects.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 3 | 2 | 2 | 1 | **-** | **-** | **-** | 2 | **-** | 3 | **-** | 1 | 2 | **-** |
| **CO 2** | 2 | 3 | 2 | 2 | 1 | **-** | **-** | **-** | 2 | **-** | 3 | **-** | 1 | 2 | **-** |
| **CO 3** | 2 | 3 | 2 | 3 | 1 | **-** | **-** | **-** | 2 | **-** | 3 | **-** | 1 | 2 | **-** |
| **CO 4** | 2 | 3 | 2 | 3 | 1 | **-** | **-** | **-** | 2 | **-** | 3 | **-** | 1 | 2 | **-** |
| **Average** | 2 | 3 | 2 | 2.5 | 1 | **-** | **-** | **-** | 2 | **-** | 3 | **-** | 1 | 2 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Software Engineering 7 Lecture Hours**

Definition of Software Engineering, S/W characteristics, applications, Software development life cycle ; Life Cycle Models – Waterfall (classical and iterative), Spiral, Prototyping & RAD Models, Software processes, Process Models – overview Agile Model and Various Agile methodologies - Scrum, XP, Lean, and Kanban. Scope of each model and their comparison in real-world case studies.

**Unit II: Requirements Modelling and Design 9 Lecture Hours**

System and software requirements; Requirements Engineering-Crucial steps; types of requirements, Functional and non-functional requirements; Domain requirements; User requirements; Elicitation and analysis of requirements; Requirements documentation – Nature of Software, Software requirements specification, Use case diagrams with guidelines, DFD, SRS Structure, SRS Case study, Design concepts and principles - Abstraction - Refinement - Modularity Cohesion coupling, Architectural design, Detailed Design Transaction Transformation, Refactoring of designs, Object-oriented Design User-Interface Design.

**Unit III: Software Reliability 9 Lecture Hours**

Introduction to Software Reliability; Hardware reliability vs. Software reliability; Reliability metrics; Failure and Faults – Prevention, Removal, Tolerance, Forecast; Dependability Concept – Failure Behavior, Characteristics, Maintenance Policy; Reliability and Availability Modeling; Reliability Evaluation Testing methods, Limits, Starvation, Coverage, Filtering; Microscopic Model of Software Risk; Classes of software reliability Models; Statistical reliability models; Reliability growth models; Defining and interpreting reliability metrics; Fault Detection and Prevention; Techniques for detecting and mitigating software faults; Static analysis tools and techniques; Dynamic analysis methods; Software Fault Tolerance; Software Maintenance and Reliability; Reliability Assessment and Evaluation; Methods for assessing and quantifying software reliability; Case Studies and Real-world Applications.

**Unit IV: Software Testing, metrics and Quality Assurance 10 Lecture Hours**

Testing types and techniques such as black box, white box, and gray box testing, functional and structural testing; Test-driven development, code coverage, and quality metrics; Testing process, design of Test cases, testing techniques - boundary value analysis - equivalence class testing - decision table testing, cause-effect graphing, path testing, data flow testing, and mutation testing. Unit, integration, system, alpha, and beta testing, debugging techniques; verification and validation techniques, levels of testing, regression testing, quality management activities, product and process quality standards (ISO9000, CMM), metrics understanding (process, product, project metrics), size metrics (LOC, Function Count, Albrecht FPA), product metrics, metrics for software maintenance, cost estimation techniques (static, single variable, multivariable models), cost-benefit evaluation techniques, Testing tools and standards such as Jira and Selenium, test automation frameworks and tools (Selenium, Appium, JUnit), performance testing and load testing, and defect management and root cause analysis.

**Unit V: Software Quality and Risk Management 10 Lecture Hours**

McCall quality factors, ISO and CMM Model, Tools and Techniques for Quality Control, Pareto Analysis, Statistical Sampling, Quality Control Charts and the seven Run Rule. Modern Quality Management, Risk Management – importance, types, process and phases, qualitative and quantitative risk analysis, Risk Analysis and Assessment, Risk Strategies, Risk Monitoring and Control, Risk Response and Evaluation. Software Reliability: Reliability Metrics, Reliability Growth Modeling. Use Case: Defect Tracking and Management. Test Automation Tools: Jira, Selenium:, Appium; JUnit.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Roger S. Pressman, "Software Engineering: A practitioner's approach", 7th Edition, McGraw Hill, 2009. 2. Pankaj Jalote, "An integrated approach to Software Engineering", 3rd Edition, Springer/Narosa, 2005. |
| **Reference books** | 1. James F. Peters, and Witold Pedrycz, "Software Engineering: an Engineering approach", John Wiley, 2007. 2. Waman S Jawadekar, "Software Engineering principles and practice", McGraw Hill, 2004. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG3015** | **Compiler Design** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 5** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** | **Formal Languages and Automata Theory-** **CSEG3055** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1.       To introduce the major concept areas of language translation and compiler design.

2.       To enrich the knowledge in various phases of compiler and its use, code optimization techniques, machine code generation, and use of symbol table.

3.       To extend the knowledge of parser by parsing LL parser and LR parser.

4.       To provide practical programming skills necessary for constructing a compiler.

**Course Outcomes**

On completion of this course, the students will be able to

**CO 1**. Comprehend different phases of compiler.

**CO 2.** Use concepts of regular grammar to build lexical analyzer.

**CO 3**. Build parsers for a context free grammar.

**CO 4**.Synthesize syntax directed translations rules.

**CO 5.** Assess code and memory optimization techniques to improve the performance of a program.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | 3 | **-** |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 2** | 2 | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 3** | 2 | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO4** | 2 | 2 | 3 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO5** | 2 | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **Average** | 2 | 2 | 3 | **-** | .6 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction 10 Lecture Hours**

Compiler, Phases and Passes, Bootstrapping, Finite State Machines and Regular Expressions and their Applications to Lexical Analysis, Implementation of Lexical Analyzers, Lexical Analyzer Generator, LEX, Formal Grammars and their Applications to Syntax Analysis, BNF Notation, Ambiguity, YACC. The Syntactic Specification of Programming Languages: Context Free Grammars, Derivation and Parse Tree, Capabilities of CFG.

**Unit II: Basic Parsing Techniques 10 Lecture Hours**

Parsers, Shift Reduce Parsing, Operator Precedence Parsing, Top Down Parsing, Predictive Parsing, Automatic Construction of Efficient Parsers: LR Parsers, The Canonical Collection of LR(0) items, Constructing SLR Parsing Tables, Constructing Canonical LR Parsing Tables, Constructing LALR Parsing Tables, Using Ambiguous Grammars, An Automatic Parser Generator, Implementation of LR Parsing Tables, Constructing LALR set of items

**Unit III: Syntax-Directed Translation 10 Lecture Hours**

Syntax Directed Translation Schemes, Implementation of Syntax Directed Translators, Intermediate Code, Postfix Notation, Parse Tree & Syntax Tree, Three Address Code, Quadruples & Triples, Translation of Assignment Statements, Boolean Expressions, Statements that alters the Flow of Control, Postfix Translation, Translation with a Top Down Parser, More about Translation: Array Reference in Arithmetic Expressions, Procedure Calls, Declaration, and Case Statements.

**Unit IV: Symbol Table 10 Lecture Hours**

Data Structures for Symbol Tables, Representing Score Information, Run Time Administration: Implementation of Simple Stack Allocation Scheme, Storage Allocation in Block Structures Language, Error Detection and Recovery: Lexical Phase Error, Syntactic Phase Errors, Semantic Phase Errors.

**Unit V: Introduction to Code Optimization 5 Lecture Hours**

Loop Optimization, the DAG Representation of Basic Blocks, Value Number and Algebraic Laws, Global Data-Flow Analysis

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, “Compilers- Principles, Techniques, and Tools”, 2nd Edition, Pearson Education Asia, 2013. 2. Robin Hunter, “The Essence of Compiler”, 2nd Edition, Pearson Publication, 2004. |
| **Reference books** | 1. Randy Allen, and Ken Kennedy, “Optimizing Compilers for Modern Architectures: A Dependence-based Approach”, Morgan Kaufmann Publishers, 2002. 2. Steven S. Muchnick, “Advanced Compiler Design and Implementation”, Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003. 3. Keith D Cooper, and Linda Torczon, “Engineering a Compiler”, Morgan Kaufmann Publishers Elsevier Science, 2004. 4. Charles N. Fischer, and Richard. J. LeBlanc, “Crafting a Compiler with C”, Pearson Education, 2008. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **EMPL004** | **EDGE – Advance Communication II** | | 1 | 0 | 0 | 0 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SFLS0006** | **Deepening Self** | |  |  |  | 2 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **PROJ3154** | **Minor Project** | | **0** | **0** | **0** | **5** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The objective of Minor Project is to develop a software application showcasing algorithm design and its subsequent implementations to solve the real-world problems.

**Course Outcomes**

On completion of this course, the students will be able to

1. Apply concepts of Data Structures, Algorithm design and Procedural Programming in the software application.
2. Use knowledge of Software engineering, computer networks, operating systems and domain of specialization to formulate and implement the problem statement.
3. Create a report capturing the entire lifecycle of project carried out in semester.
4. Develop a working software to department that meets the approved objectives and justifies the title of the project.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | 3 | 2 | 2 | **-** | **-** | 1 | 1 | 1 | 2 | 2 | 3 | 2 | **-** |
| **CO 2** | 2 | 2 | 3 | 2 | 2 | **-** | **-** | 1 | 1 | 1 | 2 | 2 | 3 | 2 | **-** |
| **CO 3** | 2 | 2 | 3 | 2 | 2 | **-** | **-** | 1 | 1 | 1 | 2 | 2 | 2 | **-** | **-** |
| **CO 4** | 2 | 2 | 3 | 2 | 2 | **-** | **-** | 1 | 1 | 1 | 2 | 2 | 2 | **-** | **-** |
| **Average** | 2 | 2 | 3 | 2 | 2 | **-** | **-** | 1 | 1 | 1 | 2 | 2 | 2.5 | 2 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

This course needs no curated course content.

# References\*

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA (Synopsis)** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 25 | 25 | 50 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Exploratory-4** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

**SEMESTER VII**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **PROJ4145** | **Capstone Project - Phase-1** | | **0** | **0** | **0** | **5** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** | **Minor Project - PROJ3154** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

As entity of university curriculum, it is mandatory for pupil to undertake a real world project. Aim of the Capstone Project is to groom pupil’s knowledge for solving technical problems through well-structured project to build holistic competency. Capstone Project -I is to be carried out by all students compulsorily to practice the theoretical concepts learnt in three years of program. The objective of Capstone Project -I is to create a software application showcasing knowledge of software engineering, software design, software architecture, domain of specialization and its subsequent implementation in any programming language. Moreover, thorough knowledge of algorithmic Efficiency and in-depth study of literature, along with acquaintance in integrated development environment in necessary.

**Course Outcomes**

On completion of this course, the students will be able to

1. Apply domain specific knowledge for software development to solve an industry oriented problems.
2. Use knowledge of software engineering, software design, software architecture and components to formulate and implement the solution.
3. Design a report capturing the entire lifecycle of project carried out in the semester.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 1 | 1 | 2 | 2 | 3 | 3 | 3 |
| **CO 2** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 1 | 1 | 2 | 2 | 3 | 3 | 3 |
| **CO 3** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 3 | 3 | 2 | 2 | 2 | 3 | 3 |
| **Average** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 2 | 1.6 | 2 | 2 | 2.6 | 3 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

This course needs no curated course content.

# References\*

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA (Synopsis)** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 25 | 25 | 50 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **SIIB4102** | **Summer Internship** | | 0 | 0 | 0 | 1 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The objective is to gain the exposure of industrial practices.

**Course Outcomes**

Students will be able to

**CO1**: Understand the industrial processes and practices.

**CO2**. Apply the knowledge to develop, manage and implement engineering solutions within the IT industry.

**CO3.** Learn and understand various skills along with professional ethics practiced by the industry.

**CO4.** Communicate and present the technical knowledge effectively.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 3 | **-** | **-** | **-** | 2 | 2 | **-** | **-** | **-** | **-** | 3 | 3 | 2 | 3 |
| **CO 2** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | **-** | 2 |
| **CO 3** | 3 | 3 | **-** | **-** | **-** | 2 | 2 | 3 | 3 | **-** | **-** | **-** | 3 | 3 | 2 |
| **CO4** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | 3 | **-** | **-** | 1 | **-** | 2 | 2 |
| **Average** | 1.5 | 2.25 | **-** | **-** | **-** | 1.5 | 1 | .75 | 1.5 | **-** | **-** | 1 | 2.25 | 1.75 | 2.25 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

This course needs no curated course content.

# References\*

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |
| --- | --- |
| Components | Report Submission/ Presentation/ Q&A |
| Weightage (%) | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Exploratory-5** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **PROJ4146** | **Capstone Project - Phase-2** | | 0 | 0 | 0 | 5 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** | **Capstone Project - Phase-1 - PROJ4145** | | **Syllabus version: 1.0** | | | |

**SEMESTER VIII**

**Course Objectives**

**Course Outcomes**

On completion of this course, the students will be able to

1. Apply domain specific knowledge for software development to solve an industry oriented problems.
2. Use knowledge of software engineering, software design, software architecture and components to formulate and implement the solution.
3. Design a report capturing the entire lifecycle of project carried out in the semester.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 1 | 1 | 2 | 2 | 3 | 3 | 3 |
| **CO 2** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 1 | 1 | 2 | 2 | 3 | 3 | 3 |
| **CO 3** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 3 | 3 | 2 | 2 | 2 | 3 | 3 |
| **CO4** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 3 | 1 | 2 | 2 | 2 | 3 | 3 |
| **Average** | 2 | 2 | 1 | 2 | 1 | **-** | **-** | 3 | 2 | 1.5 | 2 | 2 | 2.5 | 3 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

This course needs no curated course content.

# References\*

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA (Synopsis)** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 20 | 30 | 50 | 100 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Exploratory-6** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG4038** | **IT Ethical Practices** | | **3** | **0** | **0** | **3** |
| **Total Units to be Covered: 6** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version:** | | | |

**Course Objectives**

1. Define ethics and its relevance to the field of IT.
2. Identify ethical principles and frameworks applicable to IT practices.
3. Recognize and analyze ethical dilemmas in the context of AI, Cybersecurity, Secure Coding, IPR and Copyright, and Metaverse
4. Apply ethical principles to decision-making in IT projects and practices.
5. Communicate effectively about ethical issues in IT.

**Course Outcomes**

Understand the meaning of ethics and its significance in the context of information technology (IT).

1. Apply ethical principles and frameworks, such as the ACM Code of Ethics and Professional Conduct, to guide decision-making in IT scenarios.
2. Evaluate ethical dilemmas arising from emerging technologies such as Artificial Intelligence (AI), Cybersecurity, Secure Coding, IPR and Copyright, and Metaverse.
3. Integrate ethical considerations into the planning, development, implementation, and evaluation of IT projects and practices.
4. Communicate about ethical issues clearly and persuasively in IT, both orally and in writing.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PS01** | **PS02** | **PS03** |
| **CO1** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO2** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO3** | 1 | 1 | 2 | **-** | **-** | 1 | **-** | **-** | 2 | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** |
| **Average** | 1 | 1 | 2 | **-** | **-** | 1.33 | **-** | 1 | 2 | **-** | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium) 3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to IT Ethics 5 Lecture Hours**

Defining ethics and its role in IT, Ethical principles and frameworks, Ethical dilemmas in IT.

**Unit II: Ethical Considerations in Artificial Intelligence (AI) 6 Lecture Hours**

AI and its impact on society, Bias and fairness in AI, Privacy and data protection in AI, Human autonomy, and control in AI.

**Unit III: Ethical Issues in Cybersecurity 7 Lecture Hours**

Cybersecurity threats and vulnerabilities, Protecting privacy and confidentiality in cyberspace, Ethical hacking and vulnerability disclosure, Cyber warfare and cybercrime.

**Unit IV: Secure Coding Practices 7 Lecture Hours**

Software security principles and vulnerabilities, Secure coding techniques and best practices, Ethical implications of software vulnerabilities, Security testing and vulnerability assessment.

**Unit V: Intellectual Property Rights (IPR) and Copyright in IT 10 Lecture Hours**

Understanding IPR and copyright laws, Protecting intellectual property in IT, Ethical considerations in copyright and intellectual property, Open-source software, and licensing

**Unit VI: Ethical Issues in the Metaverse 10 Lecture Hours**

Defining the Metaverse and its potential impact, Privacy and security concerns in the Metaverse, Ethical considerations in avatar representation and interactions, Virtual economies and ethical dilemmas.

**Total lecture Hours 47**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Boddington, Paula. Towards a code of ethics for artificial intelligence. Cham: Springer, 2017. 2. Ethics for the Information Age by Michael J. Quinn 3. Christen, M., Gordijn, B. and Loi, M., 2020. The ethics of cybersecurity (p. 384). Springer Nature. 4. LeBlanc, D. and Howard, M., 2002. Writing secure code. Pearson Education. 5. Rockman, H.B., 2004. Intellectual property law for engineers and scientists. John Wiley & Sons. 6. Manjikian, M., 2017. Cybersecurity ethics: an introduction. Routledge. |
| **Reference books** | 1. Gold, Tandy. Ethics in IT outsourcing. CRC Press, 2012. 2. Koehn, Daryl. The ground of professional ethics. Routledge, 2006. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA(Synopsis Evaluation)** | **Mid Sem** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Specialization Tracks (Program Electives)**

1. **Artificial Intelligence and Machine Learning** **Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSAI2017P** | **Applied Machine Learning** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 07** | | **Total Contact Hours:90** | | | | |
| **Prerequisite(s):** | **Elements of AIML-**  CSAI2018 | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the core concepts and techniques of machine learning and artificial intelligence.
2. Develop machine learning models using popular libraries and frameworks.
3. Evaluate the performance of machine learning models using appropriate metrics.
4. Apply machine learning to various real-world problems and domains.

**Course Outcomes**

On completion of this course, the students will be able to

1. Recall and define key machine learning concepts, terminologies, and algorithms.
2. Describe the differences between supervised, unsupervised, and reinforcement learning.
3. Apply data preprocessing techniques to clean, transform, and prepare datasets for machine learning.
4. Apply, compare, and contrast the strengths and weaknesses of different machine learning algorithms.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 3 |
| **CO 2** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 3 |
| **CO 3** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 3 |
| **CO4** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 3 |
| **Average** | 1 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.25 | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction** **3 Lecture Hours**

Overview of machine learning and its applications, Types of machine learning: supervised, unsupervised, reinforcement, Python and libraries for machine learning (e.g., NumPy, Pandas, scikit-learn)

**Unit II: Loss functions 3 Lecture Hours**

Mean Squared Error (MSE), Mean Absolute Error (MAE), Huber Loss, Binary Cross-Entropy Loss (Log Loss), Categorical Cross-Entropy Loss, Sparse Categorical Cross-Entropy Loss, Hinge Loss (SVM Loss), Triplet Loss

**Unit III: Optimizer function 6 Lecture Hours**

Stochastic gradient descent, Mini-Batch Gradient Descent, Momentum, Adaptive gradient algorithm (Adagrad), Adam (Adaptive Moment Estimation), RMSprop (Root Mean Square Propagation), Adadelta

**Unit IV: Data Preprocessing 10 Lecture Hours**

Data Cleaning: handling Missing Data, Handling Outlier, Data Transformation: Feature Scaling, Feature Encoding, Feature Engineering, Data Reduction: Dimensionality reduction technique, feature selection, Data Splitting: Cross validation techniques, Handling imbalanced data: Oversampling techniques, under sampling techniques.

**Unit V: Regression 12 Lecture Hours**

Introduction to Regression, Regression examples, Regression models, Steps in regression analysis, Linear regression, Simple linear regression, Mathematical proof of linear regression, Least squares estimation, Least squares regression-Line of best fit, Illustration, Direct regression method, Maximum likelihood estimation, Coefficient of determination (R-squared), Checking model adequacy, Over-fitting, Detecting over-fit models: Cross validation, Logistic regression, Mathematical proof of logistic regression, multiple linear regression, Multiple linear regression model building, Mathematical proof of Multiple linear regression model, Interpretation of multiple linear regression coefficients-Partial regression coefficients, Standardized regression coefficients, Missing data, Validation of multiple regression model, regularization, ridge and lasso regularization.

**Unit VI: Classification 14 Lecture Hours**

Introduction, ML classifier, Classification and general approach, Classification algorithms, Instance based learning, K-Nearest neighbour, Decision trees, Attribute selection measure: Information gain, ID3 algorithm, Converting a tree to rules, Bayesian algorithms, Ensemble, Ensemble of classifiers, Bagging, Boosting, Random forests, Neural networks, Activation functions, Feedforward neural network, Multi-layer perceptron, Back propagation algorithm, Recurrent or feedback architecture, Perceptron rule, Multilayer networks and back propagation algorithm, Support vector machine, Classification model evaluation and selection, ROC curves, AUC curves.

**Unit VII: Clustering Techniques 12 Lecture Hours**

Introduction to Clustering, Clustering algorithms, Statistics associated with cluster analysis, General applications of clustering, Clustering as a pre-processing tool, Similarity and dissimilarity between objects, Type of data in clustering analysis, Binary variables, Nominal variables, Ordinal variables, Cluster centroid and distances, Hierarchical clustering, Hierarchical Agglomerative Clustering (HAC), Hierarchical Agglomerative Clustering: Linkage method, Hierarchical Agglomerative Clustering: Variance and Centroid method, Cluster distance measures, agglomerative clustering, Distance between two clusters, Hierarchical clustering: Time and Space requirements, K - means clustering, The K-medoids clustering method, CLARA (Clustering Large Applications), Density based clustering methods, DBSCAN.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido 2. Pattern Recognition and Machine Learning" by Christopher M. Bishop |
| **Reference books** | 1. Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy 2. Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Applied Machine Learning Lab**

**List of Experiments**

Below is a list of small machine learning-based projects suitable for this lab work. Through these projects students are expected to implement the concepts of data preprocessing and machine learning algorithms. These projects cover various machine learning techniques and can serve as valuable learning experiences:

1. Predicting Housing Prices: Develop a regression model to predict house prices based on features like location, size, and amenities.
2. Iris Flower Classification: Use the Iris dataset to build a classification model that predicts the species of iris flowers.
3. Handwritten Digit Recognition: Implement a digit recognition system using the MNIST dataset and a neural network.
4. Breast Cancer Diagnosis: Develop a breast cancer classification model using medical imaging data (e.g., mammograms).
5. Sentiment Analysis: Create a sentiment analysis tool that classifies text reviews as positive or negative using natural language processing (NLP) techniques.
6. Spam Email Detection: Build a spam email filter using text classification algorithms.
7. Predicting Stock Prices: Develop a time series prediction model to forecast stock prices.
8. Credit Risk Assessment: Build a credit scoring model to assess the creditworthiness of applicants using historical financial data.
9. Recommendation System: Create a movie or book recommendation system based on user behavior data (collaborative or content-based).
10. Anomaly Detection: Implement an anomaly detection system for detecting outliers in data (e.g., fraud detection).
11. Customer Churn Prediction: Develop a model to predict customer churn in a subscription-based business.
12. Fake News Detection: Create a model to classify news articles as real or fake based on their content.
13. Disease Diagnosis from Medical Images: Use medical imaging data (e.g., X-rays) to diagnose diseases or conditions.
14. Traffic Sign Recognition: Build a model that can recognize and classify traffic signs in images or video streams.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python", Shroff/O'Reilly, 2016. 2. Christopher M. Bishop , "Pattern Recognition and Machine Learning", Springer, 2016. |
| **Reference books** | 1. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012. 2. Sebastian Raschka and Vahid Mirjalili, "Python Machine Learning", 2nd Edition, Packt Publishing, 2017. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSAI3027P** | **Deep Learning** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Applied Machine Learning -**  CSAI7019 | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the basic concepts of neural network.
2. Understand the fundamental concepts, usage and impact of neural network, deep learning algorithms in various domains.
3. Discuss various deep-learning algorithms to solve real life problems.

**Course Outcomes**

On completion of this course, the students will be able to

1. To know the concepts of neural networks.
2. Discuss the deep learning concepts corresponding to different applications.
3. Comprehend the contemporary techniques in deep learning.
4. Analyse the concept of convolutional neural networks, recurrent neural networks, generative. deep learning and its usage.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 2** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 3** | 1 | **-** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | 1 | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **Average** | 1 | **-** | 1.25 | **-** | 0.5 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Fundamentals of Neural Networks 10 Lecture Hours**

History of Deep Learning, McCulloch Pitts Neuron, Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons, Feed Forward Neural Networks, Back propagation, Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, Singular Value Decomposition, Parameters v/s Hyper-parameters

**Unit II: The Math behind Neural Networks 12 Lecture Hours**

Computation in neural network, The matrix magic, Visualizing deep learning, The elephant in the room, Programmatic expression of deep learning’s math constructs, Operations with the tensors, Array broadcasting, Scalar product/Inner product of tensors, Morphing shapes of tensors, Gradient calculation.

**Unit III: Neural Networks and Deep Learning Basics 13 Lecture Hours**

Hebbian learning, Hebbian learning modifications: Mathematical models, Competitive learning, Error-correction learning, Boltzmann learning, Learning tasks: Pattern association, Learning tasks: Pattern recognition and function approximation, Neurons, weights, biases, transfer functions and cost/loss functions, Generalization and Overfitting, GD and Optimizers, Parameter Initialization and Hyperparameter Tuning, Data and Batch Normalization and Regularization methods, Data preparation and label preparation, Data Augmentation, CNN, Visualization of 2D convolution, Visualization of 3D convolution, Deep Neural Networks

**Unit IV: Deep Learning Model 12 Lecture Hours**

Image Classification, Image segmentation, Semantic Segmentation, Instance based segmentation, Object detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, ResNet, GoogleNet, RatinaNet,

**Unit V: Advanced Deep Learning 13 Lecture Hours**

Introduction to DL packages/ Important architectures, Deep RNNs, Recursive neural networks, Step function, Tanh function, Recurrent Neural Networks, LSTM, Generative Modelling using Deep networks, Variational Auto Encoders (VAE), Latent space, Generative Adversarial Networks (GAN’s), Transformers, Variational Auto Encoders (VAE)

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| Textbooks | 1. "Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido 2. "Pattern Recognition and Machine Learning" by Christopher M. Bishop |
| Reference books | 1. "Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy 2. "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili |
| Web Resources |  |
| Journals |  |
| MOOCs, online courses |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Deep Learning Lab**

**List of Experiments**

1. Introduction to Deep Learning: Implement a basic neural network using a deep learning framework like TensorFlow or PyTorch for a simple classification task.
2. Convolutional Neural Networks (CNNs): Build a CNN architecture for image classification on a dataset like CIFAR-10 or MNIST.
3. Transfer Learning: Fine-tune a pre-trained CNN model (e.g., VGG16, ResNet) on a custom dataset to solve a related task.
4. Recurrent Neural Networks (RNNs): Create an RNN model for sequence data like text generation or sentiment analysis.
5. Object Detection: Use a pre-trained object detection model (e.g., YOLO or Faster R-CNN) to identify objects in images or video streams.
6. Automatic Image Captioning with Keras
7. Facial Recognition
8. Digit Recognition
9. Hand Movement Recognition
10. Create synthetic datasets with the help of GANs.
11. 11 Use of transfer learning for image classification.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Introduction to Deep Learning (Neural Networks) (IBM ICE Publication). 2. Bengio Y, Goodfellow I, Courville A. Deep learning. Cambridge, MA, USA: MIT press; 2017. |
| **Reference books** | 1. Ravichandiran S. Hands-On Deep Learning Algorithms with Python: Master deep learning algorithms with extensive math by implementing them using TensorFlow. Packt Publishing Ltd; 2019 Jul 25. 2. Shanmugamani R. Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. Packt Publishing Ltd; 2018 Jan 23. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSAI3028P** | **Pattern and Visual Recognition** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Applied Machine Learning & Linear Algebra -** **CSAI2017P & MATH2059** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand fundamentals of pattern and its computational significance.
2. To study various approaches through which pattern recognition can be carried out

**Course Outcomes**

After completion of course, students would be able to:

1. Understand basic mathematical and statistical techniques commonly used in pattern recognition.
2. Understand and apply various pre-processing algorithms.
3. Apply a variety of pattern recognition algorithms.
4. Apply various algorithms for image classification.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 2** | 1 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 3** | 1 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | 1 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **Average** | 1 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

The course is aimed at specific problem, pattern recognition that is often encountered in image processing applications. Various methods that are available under statistical and learning approaches through which pattern analysis especially from recognition and detection purpose is addressed in this course.

**Unit I: Image representation and analysis 10 Lecture Hours**

Introduction to Digital Image Processing, Numerical representation of images, Image augmentation, enhancement, processing, color transforms, geometric transforms, feature recognition and extraction, Eigen values and eigen vectors, Rank of matrix and SVD

**Unit II: Feature Selection and extraction 8 Lecture Hours**

Problem statement and uses, Branch and bound algorithm, Sequential forward and backward selection, Cauchy Schwartz inequality, Feature selection criteria function: Probabilistic separability based and Interclass distance based, Feature Extraction: principles

**Unit III: Machine Learning Approaches for 20 Lecture Hours**

**Pattern Recognition**

Neural networks, how neural networks learn? Neural networks examples, Neural networks use cases, Kernel methods, Sparse kernel machines use cases, Graphical models, Mixture models and EM, Bayesian networks: Directed graphical models, Conditional probability distributions, Potential functions, Conditional independences, Sampling methods for pattern recognition, Continuous latent variables, Combining models for pattern recognition, Hidden Markov models, Markov chain Monte Carlo, The K-means algorithm, Applications of K-means.

**Unit IV: Visual Recognition 12 Lecture Hours**

Human visual recognition system, Recognition methods: Low-level modelling (e.g. features), Mid-level abstraction (e.g. segmentation), High-level reasoning (e.g. scene understanding); Detection/Segmentation methods; Context and scenes, Importance and saliency, Large-scale search and recognition, Egocentric vision, systems, Human-in-the-loop interactive systems, 3D scene understanding.

**Unit V: Recent advancements in Pattern Recognition 10 Lecture Hours**

Comparison between performance of classifiers, Basics of statistics, covariance and their properties, Data condensation, feature clustering, Data visualization, Probability density estimation, Visualization and Aggregation, FCM and soft-computing techniques, Examples of real-life datasets.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Pattern and Anomaly Detection (IBM ICE Publications) 2. Szeliski R. Computer vision: algorithms and applications. Springer Nature; 2022 Jan 3. Bishop CM. Pattern Recognition and Machine Learning by Christopher M. Bishop. Springer Science+ Business Media, LLC; 2006. |
| **Reference books** | 1. Richard O. Duda, Peter E. Hard, David G. Stork, Pattern Recognition, 2nd, Wiley, 2021. 2. Prince SJ. Computer vision: models, learning, and inference. Cambridge University Press; 2012 Jun 18. 3. Theodoridis S, Koutroumbas K. Pattern recognition. Elsevier; 2006 Apr 7. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Pattern and Visual Recognition Lab**

**List of Experiments**

**Experiment No 1** Installing Anaconda and setup up the environment, installation of supporting packages, Linear Regression

**Experiment No 2** Linear Regression and Non-Linear Regression, Linear and Non-Linear Curve fitting for regression type of problem, Error correction methods, MSE and MAE

**Experiment No 3** Logistic Regression and Sigmoid function for classification-based problem.TPR, FPR, TNR, FNR, Recall, Precision, Sensitivity, ROC-AUC curve analysis

**Experiment No 4** Un-Supervised Learning, k-Means Clustering & Elbow Rule

**Experiment No 5** Image Segmentation: Explore techniques like thresholding, region-based segmentation, and edge-based segmentation to partition an image into meaningful regions.

**Experiment No 6** Semantic Segmentation using deep learning models

**Experiment No 7** Instance segmentation using deep learning models

**Experiment No 8** Build a CNN architecture for image classification on a dataset like CIFAR-10 or MNIST.

**Experiment No 9** Isolated Random Forest Analyse and identify patterns in data to predict and take action in biological images and data to analyse DNA, RNA, and nuclei patterns to study evolution and identify patterns in gene or cell expression data to uncover new biomarkers

**Experiment No 10** Isolated Random Forest Analyse and identify patterns in data to predict and take action in biological images and data to analyse DNA, RNA, and nuclei patterns to study evolution and identify patterns in gene or cell expression data to uncover new biomarkers

**Experiment No 11** Neural network for Pattern and hand Movement Recognition

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Book provided by IBM- Data Mining & Prediction Modeling (Course code GAI06SG1 V1.0) |
| **Reference books** | 1. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, c 1995 Prentice-Hall, Inc |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG4040P** | **Computational Linguistics and Natural Language Processing** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Applied Machine Learning - CSAI2017P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. The objectives of this course are:
2. To introduce the concept of Natural Language Understanding & Natural Language Generation.
3. To develop the concept of statistical and probabilistic approach of language modelling.
4. To extend the knowledge of Large Language Model.
5. To enrich the knowledge with different corpuses and different tools being used for machine translation.
6. To provide programming skills necessary for processing natural language.

**Course Outcomes**

On completion of this course, the students will be able to

CO1: Understand the techniques in NLP.

CO2: To understand the Large Language Model

CO3: To comprehend the natural language generation.

CO4: To understand the machine translation & information retrieval techniques.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | **-** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 2** | 1 | **-** | **-** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 3** | 1 | 2 | **-** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | 1 | **-** | **-** | 3 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **Average** | 1 | .5 | **-** | 1.75 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Classical Approaches of NLP 12 Lecture Hours**

Introduction, Classical approaches to natural language processing , Understanding linguistics, Level 1: Morphology, Level 2: Syntax, Level 3: Semantics, Level 4: Pragmatics, Understanding linguistics, Traditional approach, Example: Automatic summarization using NLP, Drawbacks, Text processing, Ambiguities and computational challenges in processing various natural languages, Introduction to Real life applications of NLP such as spell and grammar checkers, information extraction, question answering, and machine translation.

**Unit II: Empirical & Statistical Approaches 12 Lecture Hours**

Corpus creation, Corpus linguistics, Types of corpora, Lexicographical implementations in corpora, Timeline of corpus linguistics, Usage areas of corpora, Traits of a good text corpus, Annotations in text corpus, NLP task-specific training corpora, Treebank annotation, Usage of annotations and corpora, Kinds of annotations, Tree banks and its construction, Need for tree bank, Types of tree bank corpus, Ambiguity in language, Segmentation, Stemming, Tokenization, Representation of word, Sentence, Word embedding, Word Senses, Linguistic Structure: Dependency Parsing. Fundamental statistical techniques, Problems of the traditional approach, how statistics helps, Problems of the traditional approach and how statistics helps, Hidden Markov model, Maximum entropy Markov model, Conditional random field model, Support vector machine, N-GRAM, Perplexity, POS Tagging, Word sense disambiguation, POS tag and Hidden Markov model, POS tagging using HMM, Viterbi algorithm, Recurrent Neural network, Vanishing Gradients and exploding gradients. Parsing, Statistical parsing, Approaches to parsing, Statistical approach, Lexicalized statistical parsing, Top-down parsing, Bottom-up parsing, Left corner parsing method, Statistical parsing: Probabilistic parser,

**Unit III: Language Modelling 12 Lecture Hours**

Word similarity and text similarity, Text similarity methods, Jaccard similarity, K-means, Cosine similarity, Word Mover’s distance, Word sense disambiguation, Complications in WSD, Methods in WSD, Evaluation of WSD, the role of language models, estimating parameters and smoothing. Evaluating language models, LSTM (Long short-term memory), GRU (Gated recurrent Unit), Part of speech tagging, BERT, XLnet, 1D-CNN for NLP, Sub-word Models, Contextual Representations, Transformers, Self-Attention for Generative Models.

**Unit IV: Machine Translation 12 Lecture Hours**

Machine translation, Rule-based machine translation, Statically Machine Translation, Neural Machine Translation, Seq2Seq Modelling, Attention, Question Answering Bot, Natural Language Generation, Neural Machine Translation, Case studies on Amazon Alexa, Google Assistant, Microsoft Cortona etc.

**Unit V: Applications of Natural Language Processing 12 Lecture Hours**

Text Summarization, Document Summarization, Sentiment Analysis, Question Answering system, Sarcasm Detection, Hostile detection Information retrieval in NLP, Image caption generation, Intelligent Tutoring System

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Christopher D. Manning, and Hinrich Schutze, “Foundations of Natural Language Processing”, 6th Edition, The MIT Press Cambridge, Massachusetts London, England, 2003. 2. Daniel Jurafsky and James H. Martin “Speech and Language Processing”, 3rd Edition, Prentice Hall, 2009. |
| **Reference books** | 1. Nitin Indurkhya, and Fred J. Damerau “Handbook of Natural Language Processing”, 2nd Edition, CRC Press, 2010. 2. James Allen, “Natural Language Understanding”, 8th Edition, Pearson Publication, 2012. 3. Christopher D. Manning and Hinrich Schütze, “Foundations of Statistical Natural Language Processing”, 2nd edition, MIT Press Cambridge, MA, 2003. 4. Hobson Lane, Cole Howard, and Hannes Hapke, “Natural language processing in action”, Manning Publications, 2019. 5. Alexander Clark, Chris Fox, and Shalom Lappin, “The Handbook of Computational Linguistics and Natural Language Processing”, Wiley-Blackwell, 2012. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Computational Linguistics and Natural Language Processing Lab**

**List of Experiments**

**EXPERIMENT-1:**

**Title: Installing various packages required for analytics in python and write code in Python for following program**

1. Write code to load CSV file containing information about employee of a company in python and draw graph showing average salary department wise.

**EXPERIMENT-2:**

**Title: Text Retrieval**

1. Connect to Twitter account and Extract first 100 tweets from it in a file.
2. Study and Implementation of Processing text (Word and Sentence Tokenization)

**EXPERIMENT-3:**

**Title: Processing Data**

1. Python code to read a text document and perform basic pre-processing techniques on the text like tokenization, stop-word-removal, lemmatization etc.
2. Study and Implementation of Morphological analysis.

**EXPERIMENT-4:**

**Title: Do text mining on extract data and Accessing text corpus**

1. Calculate word count of a given specific document and show top 10 frequent words with their frequency and Create world cloud and show graphically.
2. Study and Implementation of NER (Name Entity Recognition)

**EXPERIMENT-5:**

**Title: POS-Tagging and Tagging and Parsing**

1. Categorizing and tagging words in Twitter Data.
2. Study and implementation of POS Tagging and Chunking in a sentence.

**EXPERIMENT-6:**

**Title: Language Processor**

1. Implement N–Gram Language Mode and Smoothing.

**EXPERIMENT-7:**

**Title: Do sentimental analysis**

1. Analysis of Sentiment and Subjectivity
2. Implement sentimental analysis on IMDB Movie Reviews Dataset.

**EXPERIMENT-8:**

**Title: Do Text Summarization**

1. Analysing Meaning of Sentences
2. Implement Text Summarization on IMDB Movie Reviews Dataset.

**EXPERIMENT-9 & 10:**

**Title: Mini-Project on NLP**

Implement Mini-Project on NLP applications.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Daniel Jurafsky, and James H. Martin, “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”, 2nd Edition, Pearson, 2013. |
| **Reference books** | 1. David A. Grossman, and Ophir Frieder, “Information Retrieval: Algorithms and Heuristics”, Springer, 2004. 2. Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, “Natural Language Processing: A paninian perspective”, Prentice Hall, New Delhi, 1995. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

**2. DevOps Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDV2010P** | **DevOps Fundamentals and SCM** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Learn about the DevOps lifecycle and its stages, including continuous integration, continuous delivery, and continuous deployment.

2. Gain practical knowledge of version control systems and configuration.

3. Understand the importance of security in DevOps and learn how to integrate security practices into the CI/CD pipeline.

4. Learn the principles and techniques of demand forecasting and source code management planning to effectively match with needs.

5. Gain insights into sustainable source code management practices, to create long-term value.

**Course Outcomes**

On completion of this course, the students will be able to

CO1: Understand the principles and benefits of DevOps and its role in software development and deployment.

CO2: Develop an understanding of compliance requirements and how to incorporate them into DevOps processes.

CO3: Develop a comprehensive understanding of source code management concepts, including its objectives and components.

CO4: Understanding of management principles and techniques, including source code planning and optimization.

CO5: Gain an understanding of source code management practices and their impact on the teams' environment.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 |  | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | **-** | **-** | 0.4 |  | 2 | **-** | **-** | **-** | 0.6 | **-** | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to DevOps 12 Lecture Hours**

Definition and principles of DevOps, Benefits of DevOps adoption, DevOps culture and mindset, DevOps Lifecycle, Overview of the DevOps lifecycle stages, Continuous Integration (CI) and Continuous Delivery (CD), Automated testing and deployment, Version Control and Configuration Management, Introduction to version control systems (e.g., Git), Branching and merging strategies, Configuration management tools (e.g., Ansible, Puppet)

**Unit II: Continuous Integration and Build Automation 12 Lecture Hours**

Introduction to IaC concepts and benefits, Infrastructure provisioning tools (e.g., Terraform, CloudFormation), Managing infrastructure configurations, setting up CI/CD pipelines, Building and packaging applications, Automated testing and code quality checks, Continuous Deployment and Release Management, Strategies for releasing software updates, Managing deployment environments, Release orchestration and rollback strategies

**Unit III: Monitoring and Security 6 Lecture Hours**

Importance of monitoring in DevOps, implementing application and infrastructure monitoring, Log aggregation and analysis, Security and Compliance, DevOps security principles and practices, Implementing security controls in CI/CD pipelines, Compliance considerations in DevOps.

**Unit IV: SCM and GIT Fundamentals 12 Lecture Hours**

Overview of source code management and its role in software development, Version control systems and their benefits, Introduction to centralized and distributed SCM, Understanding repositories, commits, and revisions, Exploring the concepts of working directory, staging area, and remote repositories, Introduction to branching and tagging, Introduction to Git and its key features, Setting up a Git repository, Committing changes and viewing history, Working with remote repositories, Cloning, pushing, and pulling changes, Collaboration workflows and managing multiple contributors

**Unit V: Advanced GIT and Subversion (SVN) 1 2 Lecture Hours**

Branching strategies and best practices, merging changes, and resolving conflicts, Rebasing and cherry-picking, Git hooks and customizing workflows, managing large projects with submodules and subtrees, Git internals and understanding object storage, Overview of Subversion, and its architecture, setting up a Subversion repository, committing changes and viewing history with SVN, Branching and merging in Subversion, Repository organization and access control, Handling conflicts in Subversion

**Unit VI: SCM: Systems, Practices, and Integration 6 Lecture Hours**

Evaluating different SCM systems (Git, Subversion, Mercurial, etc.), Strengths and weaknesses of each system, Choosing the right SCM for a given project, best practices for SCM in a team environment, Integrating SCM with development workflows and tools (IDEs, build systems, etc.), Continuous integration and deployment with SCM.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Gene Kim, Kevin Behr, and George Spafford, "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win", 5th Edition, Revolution Press, 2018. 2. Jez Humble and David Farley, “Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley Professional, 2010. 3. Emma Jane, “GIT for Teams”, O'Reilly Media, 2015. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**DevOps Fundamentals and SCM LAB**

**List of Experiments**

**Experiment 1:** Creating a Local Git Repository and Executing Git common Commands on It.

**Experiment 2:** Create a GitHub Repository and Push/Pull/Clone to/from the Local repository.

**Experiment 3**: Fork a Public Repository to create a Pull Request.

**Experiment 4:** Git Revert, Git Rebase, and Git Reset Execution.

**Experiment 5:** Comparison of GitHub, GitLab, and Gutbucket SCM.

**Experiment 6:**   Create an account on AWS and create ECR.

**Experiment 7:** GitHub Actions for workflow execution.

**Experiment 8**: Create a CI/CD pipeline using scratch for deploying applications (using Git and Jenkins)

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Gene Kim, Kevin Behr, and George Spafford, "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win", 5th Edition, Revolution Press. 2. Jez Humble and David Farley, “Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley Professional, 2010. 3. “GIT for Teams” by Emma Jane, O’Reilly, 2015. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDV3024P** | **DevSecOps: Integrating security into DevOps practices** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **DevOps Fundamentals and SCM - CSDV2010P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Implement security measures within the DevOps pipeline to identify and remediate vulnerabilities effectively.
2. Foster a culture of collaboration among development, operations, and security teams to ensure continuous security monitoring and response.
3. Automate security testing and compliance checks to maintain a proactive security posture.
4. Apply best practices to secure code, infrastructure, and deployments, meeting compliance standards and minimizing risks.

**Course Outcomes**

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

**CO1**. Recall fundamental DevSecOps principles and terminology to discuss the integration of security within DevOps workflows.

**CO2.** Demonstrate the ability to implement security measures within CI/CD pipelines and IaC templates to identify and address vulnerabilities.

**CO3.** Evaluate security scan results, assess risks, and propose improvements for secure software development and deployment.

**CO4.** Design and execute a collaborative incident response plan, fostering a culture of security awareness and teamwork within DevOps environments.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 2 | **-** | **-** | 3 |  |  |  | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | 3 | **-** | **-** | 2 |  | 3 |  |  | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | 2 | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | 1 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | 0.6 | 1.4 | 0.6 | **-** | 2 | **-** | 0.6 | **-** | 0.6 | 0.4 | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Securing DevOps 12 Lecture Hours**

The DevOps Approach (Continuous integration, Continuous delivery, Infrastructure as a service, Culture, and trust), Security in DevOps, Continuous Security (Test driven security, Monitoring and responding to attacks, Assessing risks, and maturing Security).

**Unit II: Protecting Web Applications 12 Lecture Hours**

Securing and testing web Apps, Website attacks, and content security (Cross-site scripting, content security policy, cross-site request forgery, clickjacking, and Iframes protection), Methods for authenticating users (HTTP basic authentication, Password Management, Identity Providers, Sessions and cookie security, Testing Authentication), Managing Dependencies (Node.js package management, Python requirements)

**Unit III: Securing Delivery Pipeline 12 Lecture Hours**

Access control to code management infrastructure (managing permissions in GitHub Organization, Managing permissions between GitHub and CircleCI, Signing commits and Tags with GIT), Access control for container storage (Managing permissions between DockerHub and CircleCI, Signing containers with Docker content trust), Access control for infrastructure management (Managing permissions using AWS roles and policies, Distributing secrets to production system)

**Unit IV: Maturing DevOps Security: Assessing Risks 12 Lecture Hours**

What is Risk Management? The CIA triad, Establishing the top threats to an organization, Quantifying the impact of risk, Identifying threats and measuring Vulnerabilities, Rapid Risk assessment, recording and tracking risks.

**Unit V: Maturing DevOps Security: Testing and 12 Lecture Hours**

**Continuous Security**

**Testing Security:** Maintaining Security, auditing internal Applications and Services, Red teams and External Pen Testing, and Bug Bounty Programs.

**Continuous Security:** Practice and repetition: 10,000 hours of Security, Integrating Security into DevOps, Preparing for Worst, Driving the Change.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance" by Tim Mather, Subra Kumaraswamy, and Shahed Latif. 2. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg, and Andrzej Goscinski 3. Cloud Computing: Theory and Practice, by Dan C. Marinescu. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**DevSecOps: Integrating Security in DevOps Practices Lab**

**List of Experiments**

**Experiment 1:** **Setting Up a Version Control System**

* Objective: Configure and use Git for version control.
* Install Git on your system.
* Create a new Git repository.
* Add files, commit changes, and create branches.
* Push changes to a remote repository.

**Experiment 2:** **CI/CD Pipeline Setup**

* Objective: Build a basic CI/CD pipeline.
* Set up a CI server (e.g., Jenkins).
* Configure a simple pipeline that builds and deploys a sample application.

**Experiment 3**: **Security Scanning in CI**

* Objective: Implement security checks in the CI pipeline.
* Integrate a static code analysis tool (e.g., SonarQube).
* Configure the pipeline to perform security scans on code commits.

**Experiment 4:** **Infrastructure as Code (IaC)**

* Objective: Create and manage infrastructure using IaC.
* Choose an IaC tool (e.g., Terraform).
* Define infrastructure components as code.
* Provision and manage resources on a cloud provider (e.g., AWS).

**Experiment 5:** **Securing IaC**

* Objective: Apply security best practices to IaC templates.
* Implement security controls in IaC scripts.
* Scan IaC templates for security vulnerabilities.

**Experiment 6:**   **Automated Vulnerability Scanning**

* Objective: Set up automated vulnerability scanning.
* Configure a container scanning tool (e.g., Clair) in the CI/CD pipeline.
* Scan Docker images for vulnerabilities.

**Experiment 7: Security Testing and Automation**

* Objective: Automate security testing of applications.
* TIntegrate a dynamic application security testing (DAST) tool (e.g., OWASP ZAP) into the pipeline.
* Perform automated security testing on a web application.

**Experiment 8: Security Incident Response Simulation**

* Objective: Simulate a security incident and practice response.
* Create a security incident scenario (e.g., a data breach).
* Develop an incident response plan.
* Simulate and respond to the incident.

**Experiment 9:** **Compliance as Code**

* Objective: Implement compliance checks in the pipeline.
* Choose a compliance checking tool (e.g., InSpec).
* Write compliance checks as code.
* Integrate compliance checks into the CI/CD pipeline.

**Experiment 10: DevSecOps Culture and Collaboration**

* Objective: Promote collaboration between development, operations, and security teams.
* Collaborative incident response exercise involving all teams.
* Identify areas for improved communication and cooperation.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Cloud Application Development Lab Manual (IBM ICE Publications) 2. Expert AWS Development by Atul V. Mistry, Packt Publishing, March 2018 |
| **Reference books** | 1. Jenkins: The Definitive Guide: Continuous Integration for the Masses by John Smart, 1st Edition, O'Reilly Media, 2011 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDV3023P** | **Container Orchestration and Security** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **DevSecOps: Integrating Security in DevOps Practices - CSDV30242P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The Container Orchestration with Kubernetes and Security course will help the students to grasp the key skills, technology, and concepts that a Kubernetes administrator needs to know. Plan to oversee containerized workloads and administrations with industrial Organizations utilizing the Kubernetes Preparing course. It covers all the aspects, including application lifecycle management, installation, configuration and validation, networking, scheduling, security, cluster maintenance, core concepts, Azure Kubernetes, storage, and more.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Understand the significance of container over virtual machines.

**CO2:** Design the container based deployment of an application on container based platform.

**CO3:** Implement a Kubernetes Cluster and deploy a Dockerized application on a clouds based Kubernetes cluster.

**CO4:** Test and Integrate security protocols in containersed application for any vulnerability.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 2 | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | 3 | **-** | **-** | 2 |  | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | 2 | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | 1 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | 0.6 | 1.4 | 0.6 | **-** | 2 | **-** | 0.6 | **-** | 0.6 | 0.4 | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Application Containerization 12 Lecture Hours**

Understanding Containers, Challenges in the Software Industry, Problems in Software Industry Before Containers, Put that in Container! Solution by containers in the Software Industry, Virtualization, Hypervisor, Scope of Virtualization, Containers vs Virtual Machines, Understanding Containers, Containerisation Platform, Runtime and Images, Container Platform, Container Runtime, The Chroot System, FreeBSD Jails, Linux Containers (LXC), Docker

**Unit II: Introduction to Containerization and different 12 Lecture Hours**

**environments**

Docker architecture, Docker Daemon (Container Platform), Docker Rest API, Understanding Different environments: (Dev, QA and Prod), Overcoming issues with different environments, Development Environment, Testing Environment, Staging Environment, Production Environment, Virtual machines for dev/deployments, Containers for dev/deployments, Advantages and drawbacks of containerization

**Unit III: Docker Fundamentals and Internals 11 Lecture Hours**

Docker container states, docker image vs docker containers, docker image creation using docker commit & Dockerfile, Dockerfile important keywords, docker tags, persistent storage use-case, docker volumes, docker networks, creating custom networks in docker, docker registry, docker inbuilt security concepts (namespaces, cgroups)

**Unit IV: Orchestration Tools 15 Lecture Hours**

What is orchestration, Need of orchestration, Case study: Need of Orchestration , Need of Orchestration: Container and Microservices, Docker Swarm and Kubernetes, Architecture, AWS (ECS,EKS), AWS Elastic Container Services Architecture, Azure Kubernetes Services, OpenShift, KUBERNETES ON CLOUD, Monitoring of container, How to monitor

**Unit V: Container Security 10 Lecture Hours**

Docker vs Vagrant, Docker Challenges Revisited, Vulnerabilities in images (Public and Private), Denial of service attacks, Privilege escalation methods in Docker, Security misconfigurations, Container Security, Content Trust and Integrity checks, Capabilities and namespaces in Docker, Segregating Networks, Kernel Hardening using SecComp and AppArmor, Static Analysis of container(Docker) images, Dynamic Analysis of container hosts and daemons.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Liz Rice,, "Container Security Fundamental Technology Concepts that Protect Containerized Applications, O'Reilly Media, 2020. 2. José Manuel Ortega, "CandelDevOps and Containers Security and Monitoring in Docker Containers", O′Reilly, 2020. 3. Gigi Sayfan, "Mastering Kubernetes", 4th Edition, Packt Publishing, 2020. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Container Orchestration and Security Lab**

**List of Experiments**

1. Installing Vagrant & Creating basic vagrant box using VirtualBox virtualization.
2. Understanding vagrant file - Configuration - CPU, RAM, Storage, Provisioning (Shell Script).
3. Docker Machine - Installation , configuration, creating machines (on VirtualBox).
4. Docker - Installation, Configuration, Running Images.
5. Dockerfile - Containerizing application, Building Images, Tagging, Publishing.
6. Docker - Volumes, Env, Monitoring (Docker stats).
7. DTR - Docker Hub, Private Registries, Publishing images.
8. Docker Compose - Installation, Creating Compose files, Running Images using docker-compose.
9. Running Multi-Container applications using docker compose and on Swarm.
10. Kubernetes -Minikube installation and fundamentals.
11. Deploying Pods and Services on minikube.
12. Build Docker Image using .Dockerignore file.
13. Prepare and Implement Docker Container Restart Policy
14. Working with Metadata, Log File using Docker

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Application containerization by Xebia. Available at |
| **Reference books** | 1. Jarosław Krochmalski, “Developing with Docker”, Packt Publishing, 2016. 2. Adrian Mouat, “Orchestrating, clustering, and managing containers”, O'Reilly Media, Inc., 2016. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDV4014P** | **CICD Pipeline and Security** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Container Orchestration and Security-**  CSDV3023P | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The course objective of "CICD Pipeline and Security" is to provide students with a comprehensive understanding of Continuous Integration and Continuous Deployment (CICD) principles, practices, and security considerations. The course aims to equip students with the knowledge and skills to design, configure, and implement efficient and reliable CICD pipelines. It focuses on integrating security practices into the CICD process, including automated security testing, vulnerability scanning, code analysis, and secure deployment practices. The course also covers DevSecOps principles, emphasizing collaboration and the integration of security throughout the software development and operations lifecycle. Students will learn about various tools and technologies used in CICD pipelines, as well as deployment strategies, automated testing, monitoring techniques, and industry best practices. The ultimate goal is to enable students to build secure and efficient CICD pipelines that ensure the continuous delivery of high-quality software.

**Course Outcomes**

On completion of this course, the students will be able to

1. Explore the CICD toolchain, including version control, build automation, and containerization by understanding the core concepts and benefits of CICD pipelines.
2. Integrate security practices into CICD pipelines, including automated security testing and vulnerability scanning.
3. Design and implement scalable and reliable CICD workflows.
4. Apply DevSecOps principles to foster collaboration and security throughout the software development process.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 3 | 3 | **-** | 2 | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | 2 | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | 2 | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | 2 | **-** |
| **Average** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2.25 | 3 | **-** | 2 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to CICD Pipelines and Security 10 Lecture Hours**

Overview of CICD principles and benefits, Introduction to security considerations in CICD pipelines, Understanding the software development lifecycle and the role of CICD

**Unit II: Version Control and Source Code Management 10 Lecture Hours**

Introduction to version control systems (e.g., Git, SVN), Branching strategies and best practices, Integrating version control into CICD pipelines

**Unit III: Security Integration in CICD Pipelines 10 Lecture Hours**

Identifying security vulnerabilities in software development, Secure coding practices and code analysis tools, Integrating security practices into CICD pipelines

**Unit IV: Automated Testing and Quality Assurance 10 Lecture Hours**

Different types of automated testing (e.g., unit testing, integration testing), Implementing test automation in CICD pipelines, Continuous quality assurance and code quality monitoring

**Unit V: Deployment Strategies and Orchestration 10 Lecture Hours**

Overview of deployment strategies (e.g., blue-green, canary, rolling deployments), Infrastructure as Code (IaC) and configuration management, Deployment orchestration and containerization (e.g., Docker, Kubernetes)

**Unit VI: 10 Lecture Hours**

Security controls and practices in CICD pipelines, Vulnerability scanning and management, Compliance monitoring and reporting in CICD pipelines.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley, 2010. 2. Julien Vehent, "Secure DevOps: A Practical Guide to Securing Your Software Delivery Pipeline", Manning, 2018. 3. Heather Adkins, Betsy Beyer, Paul Blankinship, and Piotr Lewandowski, “Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems", O′Reilly, 2020. |
| **Reference books** | 1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, "DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", It Revolution Press, 2016. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**CICD Pipeline and Security Lab**

**List of Experiments**

**Experiment No 1** Installation of Jenkins and Execution of a simple Job in Jenkins

**Experiment No 2** Jenkins Integration with GitHub

**Experiment No 3** Jenkins Integration with GitHub and Maven

**Experiment No 4** Static Code Analysis using SonarQube

**Experiment No 5** Jenkins Integration with Sonarqube

**Experiment No 6** Create Pipeline using Jenkinsfile

**Experiment No 7** Create Pipeline using Blue Ocean Plugin

**Experiment No 8** Implementing Master/Slave Architecture in Jenkins

**Experiment No 9** Uploading Artifacts on Nexus Server using Command Line

**Experiment No 10** Nexus Integration with Jenkins

**Experiment No 11** Integration of Docker with Jenkins to generate an image of generated build

**Experiment No 12** Deployment of Docker Image on Cloud/ Local server (Nexus) using Jenkins

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley, 2010. 2. Julien Vehent, "Secure DevOps: A Practical Guide to Securing Your Software 3. Delivery Pipeline", Manning, 2018. 4. 3. Heather Adkins, Betsy Beyer, Paul Blankinship, and Piotr Lewandowski, “Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems", O′Reilly, 2020. |
| **Reference books** | 1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, "DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", It Revolution Press, 2016. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDV4013P** | **System Provisioning and Monitoring** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 09** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The course on System Provisioning and Monitoring aims to provide students with a comprehensive understanding of system provisioning methodologies, monitoring fundamentals, and relevant tools and technologies. The objective is to equip students with the knowledge and skills to automate system provisioning processes, design effective monitoring architectures, optimize system performance, and ensure security monitoring. By the end of the course, students should be able to proficiently provision systems using infrastructure-as-code techniques, implement scalable and reliable monitoring systems, identify and resolve performance bottlenecks, and incorporate security monitoring into their overall monitoring strategy. Additionally, they should be aware of emerging trends in system provisioning and monitoring, enabling them to adapt to evolving technologies and practices in the field.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Equipped with the knowledge and skills to design and implement robust monitoring architectures.

**CO2.** Effectively provision systems using automated methodologies and tools. They will understand infrastructure-as-code principles and be capable of provisioning physical and virtual servers, configuring networks, and deploying resources in a reliable and efficient manner.

**CO3.** Ability to identify and address performance bottlenecks in IT systems. They will understand techniques for capacity planning, resource allocation, load balancing, and optimization to improve system performance and ensure efficient resource utilization.

**CO4.** Integrate security monitoring practices into the overall monitoring strategy. They will understand the importance of monitoring security events, performing intrusion detection, conducting vulnerability assessments, and responding to security incidents to ensure the protection and integrity of IT systems.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 3 | 3 | **-** | 2 | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | 2 | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | 2 | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2 | 3 | **-** | 2 | **-** |
| **Average** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | **-** | **-** | 2.25 | 3 | **-** | 2 | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to System Provisioning and Monitoring 6 Lecture Hours**

Overview of system provisioning and monitoring concepts, Importance of system provisioning and monitoring in IT infrastructure management, System provisioning methodologies and best practices ,Automated provisioning tools and techniques, Infrastructure as Code (IaC) and configuration management, Provisioning physical and virtual servers, Network provisioning and configuration.

**Unit II: Monitoring Fundamentals 6 Lecture Hours**

Fundamentals of system monitoring, Types of monitoring: active monitoring, passive monitoring, and synthetic monitoring, Metrics, logs, and events: collection, analysis, and interpretation, Key performance indicators (KPIs) and service level agreements (SLAs).

**Unit III: Monitoring Tools and Technologies 6 Lecture Hours**

Monitoring tool categories: agent-based, agentless, and hybrid, Open-source and commercial monitoring tools, Infrastructure monitoring: server monitoring, network monitoring, and storage monitoring, Application monitoring: performance monitoring, availability monitoring, and error monitoring, Cloud monitoring: monitoring cloud resources and services.

**Unit IV: Monitoring Architecture and Design 6 Lecture Hours**

Monitoring architecture components: agents, collectors, dashboards, and data repositories, Scalability and high availability considerations, Data visualization and reporting, Alerting and notification mechanisms, Integrating monitoring systems with incident management and ticketing systems.

**Unit V: Performance Tuning and Optimization 4 Lecture Hours**

Identifying performance bottlenecks and issues, Performance metrics and benchmarks, Capacity planning and resource allocation, Load balancing and optimization techniques.

**Unit VI: Security Monitoring 4 Lecture Hours**

Security event monitoring and log analysis ,Intrusion detection and prevention systems (IDPS), Vulnerability scanning and assessment, Security incident response and forensic analysis.

**Unit VII: Application Performance Monitoring (APM) 4 Lecture Hours**

End-user experience monitoring, Application code profiling and performance optimization, Transaction tracing and request monitoring, Database performance monitoring.

**Unit VIII: Cloud Monitoring and DevOps 4 Lecture Hours**

Monitoring cloud infrastructure and services, Continuous monitoring in DevOps practices, Infrastructure automation and monitoring as code, Containerization and container monitoring.

**Unit IX: Emerging Trends and Future Directions 5 Lecture Hours**

Machine learning and artificial intelligence in system monitoring, Internet of Things (IoT) and monitoring challenges, Serverless computing and monitoring considerations, Proactive monitoring and predictive analytics.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Nuno Ferreira, "Provisioning, Deployment, and Operation of High-Density Cloud Services", 2. Slawek Ligus, "Effective Monitoring and Alerting: For Web Operations", O′Reilly, 2012. 3. Harjot Gill, "Cloud Monitoring and Operations", 4. John Allspaw, "The Art of Capacity Planning: Scaling Web Resources", |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**System Provisioning and Monitoring Lab**

**List of Experiments**

**Experiment :** Setting Up the Development Environment (Java, Spring Boot, IDE,

Docker, etc.)

**Experiment 1: Infrastructure as Code (IaC) Basics**

* Objective: Introduce students to IaC concepts.
* Tasks:
  + Choose an IaC tool (e.g., Terraform).
  + Write a simple IaC script to provision a virtual machine on a cloud platform.

**Experiment 2: Advanced IaC Techniques**

* Objective: Explore advanced IaC capabilities.
* Tasks:
  + Provision multiple virtual machines with interdependencies.
  + Use IaC to configure network settings and security groups.

**Experiment 3: Continuous Integration for IaC**

* Objective: Implement CI for IaC scripts.
* Tasks:
  + Set up a CI/CD pipeline (e.g., Jenkins) for IaC.
  + Automate IaC testing and deployment.

**Experiment 4: Designing Effective Monitoring**

* Objective: Teach principles of effective monitoring design.
* Tasks:
  + Identify key metrics and performance indicators for a system.
  + Develop a monitoring plan for a sample application.

**Experiment 5: Implementing Monitoring Solutions**

* Objective: Set up monitoring tools and services.
* Tasks:
  + Install and configure monitoring agents (e.g., Prometheus, Grafana).
  + Create dashboards and alerts for a monitored system.

**Experiment 6: Performance Optimization**

* Objective: Identify and address performance bottlenecks.
* Tasks:
  + Analyze system performance metrics to detect bottlenecks.
  + Implement optimizations to improve system performance.

**Experiment 7: ScaLab Exercisele Monitoring Architectures**

* Objective: Design scalable monitoring systems.
* Tasks:
  + Explore techniques for scaling monitoring infrastructure.
  + Set up distributed monitoring and data aggregation.

**Experiment 8: Security Monitoring**

* Objective: Integrate security monitoring into the overall strategy.
* Tasks:
  + Configure security monitoring tools (e.g., intrusion detection systems).
  + Create alerts for security-related events and incidents.

**Experiment 9: Incident Response Simulation**

* Objective: Simulate a security incident and practice incident response.
* Tasks:
  + Develop an incident response plan for a security scenario.
  + Execute the plan and analyze the effectiveness of the response.

**Experiment 10: Emerging Trends and Future Technologies**

* Objective: Explore emerging technologies and trends in system provisioning and monitoring.
* Tasks:
  + Research and present on a current trend or technology (e.g., containerization, serverless computing) in the field.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford, 5th Edition, Revolution Press. 2. “GIT for Teams” by Emma Jane, O’Reilly. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

**3. Cloud Computing And Virtualization Technology Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSVT2011P** | **Cloud Computing Fundamentals** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | Operating System - CSEG2060 | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To develop a strong foundation in virtualization and cloud computing, including their key concepts, principles, benefits, and distinguishing characteristics.
2. To critically evaluate the role of Hypervisors in virtualization and compare and contrast different types of virtualization, such as server, storage, and network virtualization, highlighting their unique features and use cases.
3. To analyze the diverse deployment and delivery models in cloud computing, assessing their advantages, disadvantages, and suitability for various scenarios, enabling informed decision-making for cloud adoption.
4. To assess and analyze different types of cloud workloads, considering their specific characteristics and suitability for cloud environments.

**Course Outcomes**

**CO1**. Understand the fundamental concepts and principles of virtualization and cloud computing, including their benefits and characteristics.

**CO2.** Evaluate the role of Hypervisors in virtualization and compare different types of virtualization, such as server, storage, and network virtualization.

**CO3.** Analyze the various deployment and delivery models in cloud computing and assess their advantages, disadvantages, and suitability for different scenarios.

**CO4.** Assess and analyze different types of cloud workloads considering their characteristics and suitability for cloud environments.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | **-** | 3 |
| **CO 2** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | **-** | 3 |
| **CO 3** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | **-** | 3 |
| **CO4** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | **-** | 3 |
| **Average** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Virtualization 10 Lecture Hours**

Traditional IT Infrastructures, Shortcomings of physical infrastructures, Benefits and characteristics of Virtualization, Virtual Machine (VM) Lifecycle, Comparison of traditional IT infrastructures with virtualized infrastructures, Implementing Virtualization, Triggers for virtualization, Preparation for virtualization: Server selection, Server sizing, Server criticality, Provisioning, Proximity and Locality, Transition tools for virtualization, Cost Savings, A typical hardware/software server stack, Logical Equivalence, Pre and Post Virtualization Server Stacks, Types of Virtualization: Area and technology based classification, History of Virtualization, Time-sharing systems, IBM Mainframe Virtualization, IBM PowerVM Virtualization, Extending Virtualization to x86, Hardware support for x86 Virtualization, Impact of Virtualization: Cost Impact, Manageability Impact (CAPEX and OPEX)

**Unit II: Server and Storage Virtualization 10 Lecture Hours**

Types of Server Virtualization: Process Level and system Level, Process Level: Emulation, High-Level VM, Multiprogramming, System Level: Hardware assisted Virtualization, Full Virtualization, Para Virtualization, Partial virtualization, Hybrid Virtualization; Simulation

Hypervisors:Machine Reference model,Ring levels on x86 processors, Types of Hypervisors: Type 1 and Type II, Hypervisor modules, Goldberg and Popek Criteria’s

Other Types of Virtualization: Types of storage-Directed Attached Storage(DAS), Network Attached Storage (NAS), Storage Area Network(SAN), Redundant Array of Independent Disks (RAID) and it’s levels, Storage Virtualization, its benefits and types: Host level storage virtualization, Host based mirroring, Storage level virtualization, Network based storage Virtualization; Desktop Virtualization: its working, benefits, constraints and types, Application Server Virtualization

Case Studies**:** Xen Hypervisor, VMware Workstation-Type 2 and Microsoft Hyper-V Type 1

**Unit III: Network and Application Virtualization 8 Lecture Hours**

Network Virtualization: Network virtualization overview, Virtual Private Network (VPN): VPN working, VPN types; Virtual LAN (VLAN), Advantages of VLAN; Software-Defined Networking (SDN) and Network Functions Virtualization (NFV)-Microsegmentation, Zero Trust Security Model in cloud

Operating System Level Virtualization: Programming language-level virtualization, Application-level virtualization. Application virtualization overview: Challenges in using applications in traditional install, use and update model, Solution for challenges, Architecture, Benefits of application virtualization; Containerization: application Containerization, Benefits, types, virtualization vs containerization

**Unit IV: Introduction to Cloud Computing 12 Lecture Hours**

Distributed vs Parallel computing, Elements of parallel and distributed computing, Approaches to parallel computing, Levels of parallelism.

Cloud Computing: Virtualization and Cloud, overlapping of virtualization and cloud, Areas and relative savings, Cloud definitions, Vision of cloud computing, Cloud computing value: Business and technological value, Market viewpoint, Changes for provider, Cloud and end user, Advantages and Disadvantages of cloud computing model. Distributed computing: Client server, Peer to Peer, Grid vs cloud, Cluster vs cloud, containerization and microservices in cloud evolution

Central ideas to cloud computing: Web 2.0, Utility Computing, Service Oriented Architecture, Service level Agreements Characteristics of Cloud Computing: Scalability and Elasticity, Availability and Reliability, Manageability and Interoperability, Performance & Optimization, Accessibility and Portability, Cloud Open challenges, Anatomy of a cloud: Cloud components, Cloud computing solution components, Service Catalog, User self-service portal, Service request management, Provisioning, Optimized infrastructure, Chargeback

**Unit V: Cloud Computing Service and Deployment Models 12 Lecture Hours**

Cloud Service Models**:** Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS), Pure IaaS, Pure PaaS and Pure SaaS, IaaS: IaaS features, Examples, Layers: Physical Infrastructure Layer, Infrastructure Management Software, Web Based Management Interface, Cases where IaaS suitable or not suitable, PaaS: PaaS features, Examples, Layers: Physical Infrastructure Layer, PaaS Core Middleware, Web Based Management Interface, Cases where PaaS suitable or not suitable, SaaS: SaaS features, Examples, Cases where IaaS suitable or not suitable

Case Studies**:** AWS EC2, Salesforce and social media.

Beyond Traditional IaaS, PaaS, SaaS: Serverless Computing-Function as a Service (FaaS), Database as a Service (DBaaS) and Container as a Service (CaaS), Security Considerations in Service Models: Shared Responsibility Model variation across IaaS, PaaS, SaaS, Data security & compliance (GDPR, HIPAA, SOC 2)

Cloud Deployment Models: Private cloud, Public cloud, Hybrid cloud, Community Cloud, Multi Cloud, Pros and cons of each deployment model, Cloud deployment decision factors, Time to deploy,

Public cloud: Examples, Important points about public cloud, Factor Matrix, Public cloud advantages, Public cloud disadvantages, Private cloud – Scenario, Key observations from scenario, Factor matrix, Private cloud advantages, Private cloud disadvantages, Hybrid cloud scenarios-Observations from Hybrid scenario, Factor Matrix, Hybrid cloud advantages, Hybrid cloud disadvantages

Multi cloud: Value and benefits of multicloud, Multicloud management

**Unit VI: Cloud Workloads 8 Lecture Hours**

Cloud Workloads Overview, Workload characterization, Factors that influence cloud workload, Workloads analysis hierarchy for cloud suitability considering deployment models, Types of workloads: Scientific data intensive workloads, Business and Consumer Applications, Productivity, Social Networking, Media application, Multiplayer Online Gaming; Cloud-Native Workloads & Architectures-Serverless workloads (AWS Lambda, Azure Functions), Microservices-based workloads (Kubernetes, Docker), Edge computing workloads, Security & Compliance for Cloud Workloads, Introduction to IoT, Edge and Fog Computing

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1.Buyya Vecchiola, and Selvi, “Mastering Cloud Computing”, Second Edition, McGraw Hill Education, June 2024 |
| **Reference books** | 1. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley, 2013. 2. Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah, and Kogent Learning Solutions, “Cloud Computing Black Book”, Dreamtech Press, 2014. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** | <https://nptel.ac.in/courses/106105167> |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Cloud Computing Fundamentals Lab**

**List of Experiments**

**Experiment No. 1**: Install and configure hosted hypervisor (Type II): Oracle VM VirtualBox

-Advanced Networking Configurations

-VM Performance Tuning

**Experiment No. 2:** Create a Window 7 and Ubuntu VM over Oracle VM VirtualBox and compare their performance

**Experiment No. 3:** Install and configure bare metal hypervisor (Type I): VMware ESXI

**Experiment No. 4:** Creating a virtual machine using QEMU/KVM on Ubuntu.

**Experiment No. 5:** Creating a virtual machine using QEMU/KVM on CentOS.

**Experiment No. 6**: Introduction to AWS EC2 instance and create a Linux EC2 instance and hosting a website on it.

**Experiment No. 7**: Creating a Windows EC2 instance and attaching an EBS storage to it

**Experiment No. 8**: Attaching an EBS Volume to Linux EC2 instance.

**Experiment No. 9:** Create a S3 bucket and upload an object in it.

**Experiment No. 10:** Launch Linux EC2 instance through AWS CLI

**Experiment No. 11:** Connect S3 bucket with an EC2 instance.

**Experiment No. 12:** Run a Serverless "Hello, World!" with AWS Lambda.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Mark Wilkins, “Learning Amazon Web Services (AWS): A Hands-On Guide to the Fundamentals of AWS Cloud”, First Edition, Pearson Education, 2019. 2. Buyya Vecchiola, and Selvi, “Mastering Cloud Computing”, Second Edition, McGraw Hill Education, June 2024 |
| **Reference books** | 1. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley, 2013. 2. Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah, and Kogent Learning Solutions, “Cloud Computing Black Book”, Dreamtech Press, 2014. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSVT3032P** | **Cloud Computing Architecture and Deployment Models** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Cloud Computing Fundamentals -**  **CSVT2011P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the characteristics, benefits, and limitations of cloud service models (IaaS, PaaS, SaaS) and evaluate their suitability for different organizational needs.
2. To examine the deployment models (public, private, hybrid) of cloud computing and analyze their advantages, limitations, and considerations for effective cloud infrastructure implementation.
3. To explore cloud computing reference architectures NIST, IBM & AWS and gain a comprehensive understanding of their objectives, components, and control mechanisms for designing robust cloud solutions.
4. To analyze and evaluate fundamental and advanced cloud architectures, including workload distribution, scalability, resource pooling, and redundancy, to design optimized cloud environments.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Analyze and compare the key characteristics, benefits, and limitations of IaaS, PaaS, and SaaS cloud service models to effectively assess their suitability for different organizational needs.

**CO2:** Analyze and assess the cloud deployment models (Public, Private, Hybrid) demonstrating a deep understanding of their advantages, limitations, and challenges to make informed decisions regarding the selection and management.

**CO3:** Analyze and evaluate NIST, IBM's and AWS cloud computing reference architectures to make informed decisions about their adoption and implementation in diverse usage scenarios.

**CO4:** Ability to apply fundamental as well as advanced cloud computing architectures in various business scenarios.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 3 |
| **CO 2** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 3 |
| **CO 3** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 3 |
| **CO4** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 3 |
| **Average** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Service Models (IaaS, PaaS and SaaS) 13 Lecture Hours**

Infrastructure as a Service (IaaS): Characteristics of IaaS, Comparing ISPs and IaaS, IaaS case studies, IaaS enabling technology, the trusted cloud, IaaS as the best/not best option, PaaS: Platform as a Service.

Platform as a Service: PaaS characteristics, Integrated lifecycle platforms, Anchored lifecycle platforms, Enabling technologies as a platform, Case studies: Integrated lifecycle platform, PaaS as the best/ not best option.

Software as a Service (SaaS): SaaS origin, Evolution of SaaS: Salesforce.com’s approach, Characteristics of Software as a Service (SaaS), SaaS economics and the ecosystem, Types of SaaS platforms, SaaS: Providers, Collaboration as a service, Enabling and management tools as a service, Monitoring and management tools as a service, SaaS as the best/not best option.

**Unit II: Deployment Models (Pubic, Private, Hybrid) 12 Lecture Hours**

Private Cloud Deployment: Private Cloud, Illustration of Private Cloud, Advantages of Private Cloud, Limitations of Private Cloud, Service Management, Journey into Private Cloud, Planning and Strategy, Standardization, Virtualization, Automation, Cloud, Case study – VMware vCloud, Case Study – IBM SmartCloud Entry, Private cloud.

Public Cloud Deployment:: Public Cloud, Illustration of Public Cloud, Why Public Cloud, Advantages of Public Cloud, Limitations of Public Cloud: Low degree of security and control, Lack of control on infrastructure, Configuration, Network latency and accessibility concerns, Highest long term cost; Public v/s Private: Journey into Public Cloud, Revisit the idea of adopting public cloud: Cloud vendor selection, migrating to Cloud, Cloud vendor selection, SLA – Service Level Agreements, Credits/Compensation terms, Credit process, Disaster recovery plan, Exclusions, Security and Privacy, Periodic upgrade and maintenance, Data location and Jurisdiction, Pricing and Measurability, Interoperability and Lock-in, Exit process/Termination policies, Proven track record; Public cloud vendors and Case studies: AWS, Microsoft Azure, Google GCP

Hybrid Cloud deployment: Hybrid Cloud, Why Hybrid Cloud, Illustration of Hybrid Cloud, Advantages of Hybrid Cloud, Challenges of Hybrid Cloud, Develop and manage hybrid workloads, Developing applications for hybrid cloud, Develop applications using PaaS, Managing hybrid workloads, Journey into Hybrid Cloud

OpenStack: Introduction, OpenStack Architecture, IBM SoftLayer, IBM Bluemix -Benefits of IBM Bluemix, -More Bluemix features, -Bluemix architecture.

**Unit III: Cloud Computing Reference Architectures (NIST & IBM) 10 Lecture Hours**

NIST Cloud Computing Reference Architecture (CCRA): Objectives of NIST, The conceptual reference model, Example: Usage scenarios, Cloud consumer, Cloud provider, Cloud auditor, Cloud broker, Cloud carrier, Scope of control between provider and consumer, CCRA: Architectural components, Service orchestration, Cloud service management, Business support, Provisioning and configuration Portability and interoperability, Security, Privacy, Cloud taxonomy.

IBM’s CCRA: IBM CCRA roles, Cloud service consumer, Cloud service provider, Cloud services, Infrastructure, Common Cloud Management Platform (CCMP), CCMP supports any level of virtualization, Business Support Services (BSS), Operational Support Services (OSS), Security, resilience, performance and consumption, Cloud service creator: Service development tools, IBM CCRA versions or CCRA evolution, Adoption patterns, Adoption pattern in CCRA 3.0, Examples of cloud services.

**Unit IV: Fundamental and Advanced Cloud Architectures 15 Lecture Hours**

Fundamental Cloud Architectures: Workload distribution architecture, Resource pooling architecture, Dynamic scalability architecture, Elastic resource capacity architecture, Service load balancing architecture, Cloud bursting architecture, Elastic disk provisioning architecture, Redundant storage architecture.

Advanced Cloud Architectures: Overview of the advanced cloud architecture, Hypervisor clustering architecture, Load balanced virtual server instances architecture, Non-disruptive service relocation architecture, Zero downtime architecture, Cloud balancing architecture, Resource reservation architecture, Dynamic failure detection and recovery, Bare-metal provisioning architecture, Rapid provisioning architecture, Components that can comprise the system, Automated administration pattern, Storage workload management architecture, Live VM migration.

**Unit V: Cloud Computing Reference Architecture (CCRA) – AWS 10 Lecture Hours**

What is amazon web services, Features of AWS, Web application hosting, Content and media serving, Large scale computing and huge data sets, Disaster recovery for local applications, Media sharing, Financial service grids, Time series data processing, Backup and restore to VMware cloud on AWS, Pilot light on VMware cloud on AWS, Microsoft share point on VMware cloud on AWS, Hybrid active directory trusted domain, Hybrid active directory stretched domain, Oracle RAC on VMware cloud on AWS, Batch processing, Advertisement serving, Asynchronous online gaming, Ecommerce website: Web frontend, Ecommerce website: Checkout service, Marketing and recommendations, Fault tolerance and high availability, File synchronization service, Amazon services, Amazon Simple Storage Service (S3), Amazon services developer tools, Amazon services developer tools, Amazon services security, identity and compliance, Amazon service applications.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Cloud Computing Architecture (IBM ICE Publication) 2. Cloud Computing Deployment Model (IBM ICE Publication) 3. Cloud Computing For Dummies Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper, For Dummies, Edition 1, Nov 2009. 4. Cloud Computing: Concepts, Technology & Architecture, Thomas Erl,Pearson Education India, Jan 2014 5. OpenStack Essentials, Dan Radez, PackIT publication, publications , 2nd Edition, Jan 2016 |
| **Reference books** | 1. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley, 2013. 2. Gautam Shroff, “Enterprise Cloud Computing Technology Architecture Applications”, Cambridge, 2010. 3. , “About Openstack: A Comprehensive Tutorial To Revolutionize Cloud Computing Solutions”, Independently published, March 27, 2023. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Cloud Computing Architecture and Deployment Models Lab**

**List of Experiments**

**Experiment-1: AWS EC2 (IaaS)**  
1.1. Creation of Windows instance on AWS EC2 and connecting to it through RDP client and .pem key pair.

1.2 Creation of Linux Instance on AWS EC2 and connecting it to through .ppk keypair using PuTTY.

**Experiment-2: Azure & GCP (IaaS)**

2.1. Create Windows and Ubuntu machine Virtual Machine in Azure and connect to it.

2.2. Create a Windows Virtual machine instance in GCP and connect to it.

**Experiment-3: AWS Autoscaling and Load Balancers (IaaS)**

* 1. Autoscaling can be performed on EC2 instances (Scale manually & Scale on demand)
  2. Use Application Load Balancer in AWS

**Experiment-4: AWS Storage (IaaS)**  
4.1. EBS Volume creation and attaching it to EC2 instance (Windows as well as Linux instances)

* 1. Creating a S3 bucket

**Experiment-5: AWS Storage (IaaS)**

* 1. Changing S3 bucket access permissions, applying replication and lifecycle rules.

5.2. Host a static website on S3

**Experiment-6: Identity and Access Management /Security in AWS Cloud Environment**  
6.1. Create user, role and policies and access it through Management console and CLI.

6.2. Show how to use MFA in IAM

6.3. Adding a WAF to a Web application & its Load balancer and blocking access

**Experiment-7: Serverless Computing (AWS Lambda)**  
7.1. Creating, starting and stopping EC2 instances using AWS Lambda.

**Experiment-8: AWS ElasticBeanstalk (PaaS)**

8.1 Create and manage a LAMP stack using Amazon EC2 Instance and host an html file on it (Iaas+PaaS).  
8.1. Deploy an application on Elastic Beanstalk

**Experiment-9: Openstack Installation and VM deployment**

9.1. Learn Hands-on OpenStack Installation: Install OpenStack on single node using Packstack

9.2. Familiarize with OpenStack dashboard – Horizon: Go through each and every component of the dashboard and understand the meaning of the sections and components therein

9.3. Deploy a virtual machine instance [using OpenStack Nova]: Create, Deploy, Test and Destroy a VM instance using OpenStack Nova component via Horizon

**Experiment-10: Openstack**

**10.1.** Deploy a Linux VM from an ISO image [OpenStack – Nova, Cinder and Glance]: Create an image in OpenStack Glance, Create a VM with a volume and deploy the image

10.2. Deploy a VM from an image snapshot [OpenStack – Nova, Cinder and Glance]: Create a VM, take a snapshot of the VM and store the VM image in Glance and create a new VM image in Nova from the created snapshot

**10.3.** User and Project Management [OpenStack – Horizon advanced, Keystone]: Explore PM functions in OpenStack and try basic settings

**Experiment-11: Openstack and components**

**11.1.** Common cloud management tasks [OpenStack – Horizon, Keystone]  
Work with quotas in a Project – assign, increase, decrease quotas for compute and storage

**11.2.** Adding a new compute node [OpenStack – Nova advanced]  
Add a new compute note in Nova and create VMs on the new new node

**Experiment-12: Overview of Nagios**   
**12.1** Explore monitoring tool Nagios and check the parameters it can monitor

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. IBM Material Cloud Deployment Model Lab 2. Theo H King, “The Ultimate Guide From Beginners To Advanced For The Amazon Web Services”, Independently Published, 2020. 3. Kevin Jackson, Cody Bunch, and Egle Sigler, “OpenStack Cloud Computing Cookbook”, 3rd Edition, Packt, 2015. |
| **Reference books** | 1. , and , “AWS Cookbook”, 1st Edition, O'Reilly Media, 2021. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDV4016P** | **Containerization and DevOps** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Cloud Computing Architecture and Deployment Models -** CSVT3032P | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The objective of this course are

1. To understand Containerization with Docker, emphasizing automation and orchestration.

2. To master DevOps fundamentals for collaborative and automated software delivery.

3. To gain practical knowledge of DevOps applications in real-world scenarios.

4. To explore DevOps principles, lifecycles, and essential tools for efficient software development and deployment.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Comprehensive understanding of containerization and the principles of DevOps.

**CO2:** Apply Docker fundamentals by creating and managing Docker containers, demonstrating comprehension and application of containerization concepts.

**CO3:** Demonstrate a deep understanding of container orchestration and apply it practically by creating and managing services using container orchestration platforms.

**CO4:** Assess DevOps adoption strategies, analyze its impact on business, and synthesize cross-functional team dynamics and tool selection within the DevOps context.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | 1 | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | 3 |
| **CO 2** | **-** | 1 | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 1 | 3 |
| **CO 3** | **-** | 1 | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 1 | 3 |
| **CO4** | **-** | 1 | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 1 | 3 |
| **Average** | **-** | 1 | 0.75 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1.25 | 0.75 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Containerization and DevOps 8 Lecture Hours**

Virtualization and its types-Server Virtualization, Operating System Virtualization

**Containers & DevOps:** Introduction to Containers, Understanding DevOps Principles and Practices, Benefits of Containerization in DevOps, Docker vs. Virtualization, Different vendors for containers in the market,

**Docker:** Introduction to Docker: What is Docker and What isn’t Docker?, Overview of Docker editions, Installation of the Docker engine,Docker terminology, Docker community, Docker community edition, Docker enterprise edition, Build Kit features, Docker compose, Docker architecture-Docker host, Docker Daemon, Docker Hub, Docker API, Docker Objects, Docker and DevOps tools

**Software Designing Architectural Approaches:** Server based and Serverless architecture-Monolithic, Service Oriented Architecture, Microservices, Comparison of different architectural approaches, Docker and Microservices-apart and together

**Unit II: Docker Fundamentals 12 Lecture Hours**

**Basic Commands**: Docker container Lifecycle commands, Docker desktop , Checking docker version, Running your first NGINX application, Docker repository, Docker tags, Docker TAG examples, Docker TAG scenario, Tagging scheme, Docker images, Layers, Docker file, Docker file instructions, Managing containers and images, Creating Docker Images, Dockerfile Best Practices, Pushing Images to Docker Hub, Running your own Docker container.h

**Data Management in Docker:** Persisting data in docker, Approaches: Volumes, bind mount, Differences, Volumes-creation, listing, --mount flag, -v flag, removing, inspecting bind mount- creation, --mount flag, -v flag, removing, tmpfs mount, Use case scenarios volumes, bind mount and tmpfs mount.

**Networking in Docker:** Networking in Docker, Docker network drivers- bridge, host, overlay

ipvlan, macvlan, Publishing ports, IP address and host name, DNS services, Network Drivers use case summary, creating and removing a user-defined bridge, managing a user-defined bridge- connect a container to a user defined bridge, Disconnect a container from a user-defined bridge, Connect a container to the default bridge network.

**Continuous Integration**: Docker as a build environment, GitHub Actions, Building CI/CD Pipleine using Git Hub actions.

**Unit III: Automation and Orchestration 12 Lecture Hours**

**Container Orchestration need and Overview:** Key concepts in orchestration, popular orchestra platforms: Swarm Docker, Kubernetes, Apache mesos

**Docker Compose:** Features, use cases, history and using Docker Compose

**Docker Swarm:** Feature highlights, Swarm mode key concepts-swarm, nodes, services & tasks, load balancing; Create a swarm, Add nodes to swarm, Deploy a service, Inspect the service, Scale the service, Delete the service, Apply rolling updates, Drain a node, Use swarm mode routing mesh

**Kubernetes:** Overview, Traditional deployment era, virtualized deployment era , container deployment era, Need of Kubernetes, what Kubernetes is not?, Kubernetes components: control plane components, Node components, Addons, Kubernetes API, Cluster Architecture-Nodes, Communication between nodes and the control Plane, Controllers, Leases, Cloud Controller Manager, Container runtime Interface, Garbage Collection, Containers: Images, Container Environment, Runtime class, Container Lifecycle hooks, Workloads- Pods and Workload Resources, Services, Load balancing and Networking, Storage, Configuration, Policies and security

**Case Study:** Docker Swarm vs Kubernetes

**Case Study:** Amazon ECS and EKS services

**Unit IV: DevOps: Principles and Practices 8 Lecture Hours**

Overview, Working, Benefits, DevOps history, DevOps principles and lifecycle, DevOps practice, DevOps adoption: Deming, lean manufacturing, and Kaizen, Lean manufacturing, Lean standards of manufacturing, DevOps: IBM view, Four DevOps adoption paths- Steer adoption path, Develop and test adoption path, Collaborative development, Continuous testing, Way to deployment, Continuous customer feedback and optimization,

DevOps architecture and resilience, Cloud resiliency, DevOps resiliency, Four stages of the resilience process-Detect, DevOps style; Alert with a cloud and DevOps mindset; Respond & recover using automation and appropriate failover strategy; Refine & test, achieving incremental improvements

**Unit V: DevOps Adoption and Business Patterns 8 Lecture Hours**

Business needs for DevOps , DevOps teams and cross functioning of teams, Silos in the world of software development and their role in project delivery, DevOps teams and cross functioning of teams, Application team v/s. Platform team, System admins and other stakeholders, Continuous integration vs continuous deployment vs continuous delivery, DevOps tools- Continuous development, Continuous integration, Continuous Testing, Continuous Deployment, Continuous Monitoring, Lean Thinking and Methods – Kaizen, Agile Vs DevOps, DevOps impact on developers, DevOps impact on operations, Successful DevOps adoption, Challenges of DevOps adoption, Introduction to Kanban, Types of kanban board, create a kanban board, Kanban with IBM tools. Scrum application delivery pipeline and support team, an orchestration framework for continuous delivery, Software release plan, Feedback and learning from feedback and improving the delivery, DevOps toolchain, DevSecOps, DevOps vs SRE. Select the right tool for DevOps: Docker, Kubernetes, Puppet, Ansible, Other tools, DevOps monitoring tools, Version control, and code repository

**Unit VI: DevOps Tools 12 Lecture Hours**

**Version Control Tools:** GitHub-GitLab-BitBucket, GitHub-CLI, Desktop, Branches, forks, and Pull requests, Repositories, GitHub actions, GitHub packages, Webhooks, API

**Continuous Integration Tools:** Jenkins, TravisCI, Jenkins- Introduction to Jenkins, Jenkins architecture and components: Installation and setup of Jenkins, Creating and Configuring Jenkins Jobs Jenkins Pipelines Source Code Management (SCM) Integration Building and Testing with Jenkins

**Infrastructure as Code (IaC) Tools:** Understanding IaC, Popular IaC tools-Terraform, Ansible

**Monitoring and Logging Tools:** Monitoring and logging in DevOps**,** Popular monitoring and logging tools- Prometheus, ELK stack

**Collaboration in a DevOps Team:** Popular collaboration and communication tools- Slack, Microsoft Teams, Integrating tools into the DevOps workflow

**Case study:** CI/CD technique.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Container Orchestration & Infrastructure Automation, IBM ICE Publications. 2. James Turnbull, “The Docker Book: Containerization is the new virtualization”, First Edition, Shroff Publishers, 2019. 3. Jason Cannon, “Docker: A Project-Based Approach to Learning”, Independently published, 2021. 4. Deepak Gaikwad, Viral Thakkar, “DevOps Tools from Practitioner's Viewpoint”, Wiley, 2019. |
| **Reference books** | 1. Ian Miell and Aidan Hobson Sayers, “Docker in Practice”, 2nd Edition , Manning Publications, 2019. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Containerization and DevOps Lab**

**List of Experiments**

**Experiment 1:** Introduction to Dockers

1. Setup & Running a Container
2. Performance evaluation of Virtual box machine deployment and Dockers containers deployment.

**Experiment 2:** Docker Installation, configuration, Running Images

**Experiment 3:** Deploying web applications with Docker

**Experiment 4:** Dockerfile - Containerizing application, Building Images, Tagging, Publishing

**Experiment 5**: Docker - Volumes, Env, Monitoring (Docker stats), Docker Networks

**Experiment 6:** Docker Compose - Installation, Creating Compose files, Running Images using Docker-compose

**Experiment 7:** Create a CI CD pipeline for deploying web application using jenkins

**Experiment 8:** Ansible

**Experiment 9**: Chef

**Experiment 10:** Install, setup and run SonarQube for a local scan.

**Experiment 11**: Orchestration using Docker compose on multi container applications.

**Experiment 12:** Study and Analyse container orchestration using Kubernetes.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. IBM ICE Publications 2. The Docker Book: Containerization is the new virtualization by James Turnbull, Shroff Publishers, First Edition, 2019 3. Docker: A Project-Based Approach to Learning by Jason Cannon, **September 2021** 4. DevOps Tools from Practitioner's Viewpoint, by Deepak Gaikwad, Viral Thakkar, Wiley, Jan 2019 |
| **Reference books** | 1. Ian Miell, and Aidan Hobson Sayers, “Docker in Practice”, 2nd Edition , Manning Publications, 2019. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSVT4022P** | **Cloud Application Development** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Containerization and DevOps -** **CSDV4016P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Student should be able to get solid theoretical background of high computing paradigms.

2.Students get adequate conceptual understanding of cloud application development platform and framework.

3. Student can explore cloud applications and identify applications suitable for implementation on cloud.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Correlate the underlying technologies for cloud application development with basic programming skills

**CO2:** Apply the concepts of high throughput, data intensive applications and Task programming

**CO3:** Discover various cloud computing platforms for application development

**CO4:** Analyse the use of automation in cloud application development and understand the real time use cases

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 2 | 3 |
| **CO 2** | 1 | 1 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 2 | 3 |
| **CO 3** | 1 | 1 | 2 | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 2 | 3 |
| **CO4** | 1 | 1 | 2 | 1 |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 3 |
| **Average** | 1 | 0.6 | 2 | 0.5 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 0.75 | 1 | 2 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Basics of Cloud Computing and introduction 12 Lecture Hours**

**to MapReduce programming for application development**

Business case for implementing cloud application, Requirements collection for cloud application development, Cloud service models and deployment models, Open challenges in Cloud Computing: Cloud inter-operability and standards, scalability and fault tolerance, security, trust and privacy, Data intensive computing: MapReduce programming, what is data-intensive computing? Characterizing data-intensive computations, Data intensive research issues, Historical perspective, The early age: High-speed wide area networking, The early age, Data grids, Data clouds and big data,

**Unit II: Understanding basic Task Programming skills 12 Lecture Hours**

**and MapReduce Programming**

Understanding the basics of HTML, javascript, Client-side web scripting: DOM and AJAX, Server-side applications: node.js, Introduction to task programming, High-throughput computing: Task programming, Task computing, Task-based application models, MPI reference scenario, MPI program structure, Workflow applications with task dependencies, Workflow technologies abstract model of workflow system, Cloud application platform task-based programming, Task programming model scenario, iTask and cloud application platform task, iTask interface, Wrapping an iTask into an cloud application platform instance, Controlling task execution, Dynamic task submission. Databases and data-intensive computing, Technologies for data intensive computing, High performance distributed file systems and storage clouds, Not only SQL (NoSQL) systems, Prominent implementations supporting data: Intensive applications, Amazon dynamo architecture, Google bigtable architecture, Apache Cassandra, Hadoop HBase, Programming platforms, The MapReduce programming model, MapReduce computation workflow, Google MapReduce infrastructure overview, Variations and extensions of MapReduce

**Unit III: Application Development: Google and Microsoft Azure 12 Lecture Hours**

Accessing the clouds: Web application vs Cloud Application, Frameworks: Model View Controller (MVC), Struts, Spring, Maven, Gradle. Cloud platforms in Industry – Google AppEngine (serverless platform for build scalable web and mobile back ends in any programming language), Microsoft Azure, Openshift (building containerized applications), CloudFoundry (provides a highly efficient, modern model for cloud native application delivery on top of Kubernetes)

**Unit IV: Application Development on AWS and DevOps tools 12 Lecture Hours**

Best practices in architecture cloud applications in AWS cloud, Elastic Beanstalk (Deploy and Provision web applications), Amazon Simple Queue Service (SQS), RabbitMQ, Cloud applications: Amazon Simple Notification Service (Amazon SNS), multi-player online game hosting on cloud resources, building content delivery networks using clouds. Puppet and Chef – steps for automation: Introduction, files and packages, services and subscriptions, exec and notify, facts, conditional statements and logging, configuration management tools, automation, configuration orchestration

**Unit V: APIs for cloud application development and 12 Lecture Hours**

**real time use cases**

RESTful API Usage, RESTful API Design & Implementation, Add Functionality to REST API, storing objects in the Cloud, Session management, Working with third party APIs: Overview of interconnectivity in Cloud ecosystems. Facebook API, Twitter API, Google API. Use cases of cloud applications, Scientific applications, Healthcare: ECG analysis in the cloud, Biology: Protein structure prediction, Gene expression data analysis for cancer diagnosis, GeoScience : Satellite image processing, Business and consumer applications, SalesForce.com, Productivity, DropBox, Google Docs, EyeOS, Social networking : Facebook, Media applications: animoto, 3D rendering on private clouds, Video encoding on the cloud: Encoding.com, Multiplayer online gaming scalable processing of logs for network games.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. David E. Y. Sarna, “Implementing and Developing Cloud Computing Applications”, CRC Press, 2010. 2. Shagun Bakliwal, “Hands-on Application Development using Spring Boot: Building Modern Cloud Native Applications by Learning RESTFul API, Microservices, CRUD Operations, Unit Testing, and Deployment”, 1st Edition, BPB Publications, 2021. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Cloud Application Development Lab**

**List of Experiments**

**Experiment 1:** Installation of Single Node Hadoop Cluster on Ubuntu and run simple program like wordcount.

**Experiment 2:** Install Google App Engine. Create Hello World App and other Simple Web Applications.

**Experiment 3**: Build Apps at Scale with Google App Engine.

**Experiment 4:** Setup and Configure Cloud application development platform (Like Amazon application development platform).

**Experiment 5:** Using cloud Platform API develop and deploy prototype application.

**Experiment 6:**   Generate high volume workloads of applications on cloud platform

**Experiment 7:** Data Intensive Computing: Map-Reduce Programming implement a sample Map-Reduce procedure using cloud platform

**Experiment 8:** Developing Message-Based Applications with RabbitMQ

**Experiment 9:** Implement data analytics for some application using cloud platform

**Experiment 10:** Implement Machine Learning model/s on cloud platform like Sage Maker, Amazon Polly etc.

**Experiment 11:** Install apache http server, enable and start the services using Chef Cookbook.

**Experiment 12**: Create a CI/CD pipeline using scratch for deploying application (using Git and Jenkins)

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Cloud Application Development Lab Manual (IBM ICE Publications) 2. Atul V. Mistry, “Expert AWS Development”, Packt Publishing, 2018. |
| **Reference books** | 1. John Smart, “Jenkins: The Definitive Guide: Continuous Integration for the Masses”, 1st Edition, O'Reilly Media, 2011. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSVT4024P** | **Cloud Computing Security and Management** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 06** | | **Total Contact Hours:75** | | | | |
| **Prerequisite(s):** | **Cloud Application Development - CSVT4018P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Develop a comprehensive understanding of cloud security fundamentals,
2. Analyze the importance of data security in the cloud and its principles,
3. Evaluate the role of IAM in cloud security and its components, goals, and principles.
4. Apply network security fundamentals and CIA principles in cloud environments.
5. Apply secure coding practices, including input validation, authentication, and error handling in cloud applications.

**Course Outcomes**

**CO1:** Assess the security challenges and vulnerabilities specific to cloud infrastructure, including physical security measures and virtualization technology

**CO2:** Identify and define key data security concepts and mechanisms used to protect data in transit, at rest, and during processing in cloud environments.

**CO3**: Implement and assess IAM components and policies in cloud environments,

**CO4**: Evaluate the importance of network and application security in cloud computing, demonstrating an understanding of security fundamentals

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 1 | 3 |
| **CO 2** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 1 | 3 |
| **CO 3** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 1 | 3 |
| **CO4** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 1 | 3 |
| **Average** | 1 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 1 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Security Overview 8 Lecture Hours**

Overview of cloud security- Security Services - Confidentiality, Integrity, Authentication, Nonrepudiation, Access Control - Basic of cryptography - Conventional and public-key cryptography, hash functions, authentication, and digital signatures.

Common security risks in cloud environments, Threat modeling in the cloud, Security considerations for different cloud deployment models, Cloud security: Physical Security, Network Security, Data Security, Identity and Access Management (IAM), Application Security, Virtualization Security, Operating System Security, Security Compliance and Governance, Cloud Access Security Broker (CASB) , Logging and Monitoring, Incident Response and Forensics, Encryption Key Management, Multi-Factor Authentication (MFA), Backup and Recovery, Security Training and Awareness, Zero Trust Security, Shared responsibility model, Cloud Security Alliance

**Unit II: Infrastructure security & Virtualization Security 12 Lecture Hours**

Introduction to Physical and Infrastructure Security in the Cloud: Role of physical security in cloud computing, Key physical security components and objectives, Cloud infrastructure and data center architecture, Security challenges in shared data center environments, Data center architecture and design considerations, Redundancy and fault tolerance, Environmental controls (temperature, humidity, fire suppression), Physical access control and monitoring, Intrusion detection systems (IDS) and intrusion prevention systems (IPS), Video surveillance and access control systems, Visitor management and access logging, Data center personnel and roles (e.g., NOC, SOC), Security awareness and training for data center staff, Change management and incident response in data center operations, Asset management and tracking Data replication and backup strategies, Failover and high availability configurations, Testing and simulating disaster scenarios

Virtualization security threats and vulnerabilities, Isolation and segmentation of VMs, VM escape attacks and mitigation, Security for containerized applications

OS-Level Security: Hardening operating systems in the cloud, Patch management and vulnerability assessment, Anti-malware and intrusion detection for OS instances, OS-level access controls and permissions Secure OS Configuration: Secure boot and integrity verification, Network security configurations for OS instances, Secure shell (SSH) and remote access security, Application and service hardening OS Security Best Practices: OS security baseline standards, Security updates and patch management policies, Security information and event management (SIEM) integration, Monitoring and auditing OS-level activities

Case Studies: VM escape attack mitigation in a cloud service provider environment, OS-level security in a containerized application deployment

**Unit III: Data Security 7 Lecture Hours**

Importance of data security in the cloud, Data security principles: CIA (Confidentiality, Integrity, Availability). Shared responsibility model in cloud security,

Data Classification and Protection: Data classification and sensitivity assessment, Data protection requirements (e.g., PII, PHI), Data masking and anonymization, Data retention policies and disposal Encryption Techniques: Encryption fundamentals, Data encryption at rest using disk and database encryption, Encryption in transit with TLS/SSL, Encryption for data during processing (homomorphic encryption) Compliance and Regulatory Requirements: Secure Data Sharing and Collaboration, Secure file sharing and collaboration tools, Secure data exchange with external parties, Data leakage prevention (DLP) and protection mechanisms, Implementing secure data sharing policies Data Security in Multi-Cloud Environments: Challenges and considerations for multi-cloud data security, Data replication and synchronization, Data governance in multi-cloud architectures, Data compliance frameworks (e.g., GDPR, HIPAA), Auditing and monitoring for compliance, Cloud provider compliance certifications, Privacy and data protection in the cloud

**Unit IV: Network Security 7 Lecture Hours**

Importance of network security in the cloud, Network security fundamentals: CIA (Confidentiality, Integrity, Availability), Network security in a shared responsibility model

Cloud Network Architecture and Components: Cloud network models (VPC, VNet), Subnetting and IP addressing in the cloud, Load balancers and content delivery networks (CDNs), Virtual private clouds (VPCs) and security groups Network Protocols and Encryption: Transport layer security (TLS/SSL), VPNs and encryption for data in transit, Secure tunneling protocols (IPsec), Securing DNS and BGP Network Security Controls and Policies: Firewall fundamentals and best practices, Intrusion detection and prevention systems (IDS/IPS), Network access control lists (NACLs) and security policies, Security groups and network segmentation Secure Data Transmission and VPNs: Site-to-site VPNs for hybrid cloud environments, Remote access VPNs and secure connectivity, VPN protocols (PPTP, L2TP, OpenVPN), VPN client configuration and security considerations Network Monitoring and Security Analytics: Network monitoring tools and techniques, Real-time traffic analysis and anomaly detection, Logging and alerting for network security incidents, Security information and event management (SIEM) integration Cloud Network Security Best Practices: Implementing network security policies and procedures, Security groups and firewall rule management, Network security automation and orchestration, Network security in serverless and containerized environments

Case Studies: DDoS mitigation in a cloud-hosted e-commerce platform, Securing microservices communication in a cloud-native applications

**Unit V: Identity and Access Management (IAM) in the Cloud 6 Lecture Hours**

The role of IAM in cloud security, IAM components and stakeholders, IAM goals and principles, IAM trends and challenges

Authentication and authorization in the cloud, Authorization models (RBAC, ABAC), Role-based access control (RBAC), Identity federation and single sign-on (SSO), IAM Technologies and Protocols: LDAP and Active Directory, OAuth 2.0 and OpenID Connect, SAML (Security Assertion Markup Language), OAuth and API security, Multi-factor authentication (MFA), IAM Policy and Governance: Creating IAM policies and procedures, IAM policy lifecycle management, Policy enforcement and compliance, IAM auditing and reporting

Case Studies: AWS IAM and Azure AD

**Unit VI: Application Security 5 Lecture Hours**

Importance of application security in the cloud, Application security fundamentals, OWASP Top Ten and common vulnerabilities

Secure Coding Practices: Secure coding principles and best practices, Input validation and output encoding, Authentication and authorization in cloud applications, Error handling and logging for security Data Security in Cloud Applications: Data protection and encryption, Handling sensitive data (PII, PHI), Data access controls and role-based access control (RBAC), Data leakage prevention (DLP) in applications API Security: API security considerations and challenges, OAuth 2.0 and OpenID Connect for authentication and authorization, Rate limiting and throttling, Securing API endpoints and data transmission Application Security Testing: Static application security testing (SAST), Dynamic application security testing (DAST), Interactive application security testing (IAST), Continuous integration/continuous deployment (CI/CD) pipeline security testing Cloud Application Security Best Practices: Secure application development life cycle (SDLC), Security by design in cloud-native applications, Web application firewalls (WAFs) and application security policies, Secure application deployment strategies

Case Studies: Securing a SaaS-based customer portal, API security challenges in a cloud-native fintech application

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Ronald L. Krutz, and Russell Dean Vines, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", Wiley Publishing, 2010. 2. Tim Mather, SubraKumaraswamy, and ShahedLatif, “Cloud Security and Privacy", O’Reilly Media, Inc., 2009. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Cloud Computing Security and Management Lab**

**List of Experiments**

**Experiment 1:** Building your first Virtual Private Cloud-AWS

**Experiment 2:** Delegate access across AWS accounts using roles-AWS

**Experiment 3:** Create a customer managed policy-AWS

**Experiment 4:** Use attribute-based access control (ABAC)-AWS

**Experiment 5:** Permit users to manage their credentials and MFA settings (AWS)

**Experiment 6**: Implement Azure key vault

**Experiment 7:** Manage Access to Azure with Role-Based Access Control

**Experiment 8:** Manage Identity and Access- MFA, and Conditional Access (Azure)

**Experiment 9:** Manage Identity and Access- Implement Directory Synchronization (Azure)

**Experiment 10:** Smart Access Control on GCP: Web, programmatic, and command-line access

**Experiment 11:** Write an IAM policy by using client libraries (GCP)

**Experiment 12:** Using AWS WAF to control access to your content (AWS)

**Total Lab hours 30**

|  |  |
| --- | --- |
| **Textbooks** | 1. Ronald L. Krutz and Russell Dean Vines, “Cloud Security: A Comprehensive Guide to Secure Cloud Computing”, Wiley Publishing Inc., 2010. 2. Tim Mather, Subra Kumaraswamy and Shahed Latif, “Cloud Security and Privacy”, O’Reilly Media Inc, 2009. 3. Imad M. Abbadi, “Cloud Management and Security”, Wiley Publishing Inc., 2015. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

**4 . FULL STACK DEVELOPMENT Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSFS2004P** | **Frontend Development** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the fundamentals of web development and front-end technologies.

2. Design and develop responsive web pages using HTML and CSS.

3. Build interactive web applications using JavaScript.

4. Utilize front-end frameworks and libraries to streamline development processes.

5. Implement best practices for accessibility and usability in web design.

6. Optimize web performance and user experience.

7. Collaborate effectively with backend developers and designers.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Understand Web Development Fundamentals

**CO2:** Create and build web pages and applications.

**CO3:** Utilize front-end frameworks and optimize web performance.

**CO4:** Apply security best practices and conduct testing.

**CO5**: Demonstrate project development skills.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | **-** | **-** | 0.4 | **-** | 2 | **-** | **-** | **-** | 0.6 | **-** | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Frontend Development, Web Design 12 Lecture Hours**

Overview of web development and frontend technologies, Introduction to development environments (IDEs, text editors, browser developer tools), HTML5 essentials: tags, attributes, semantic markup, CSS fundamentals: selectors, box model, layout techniques, Introduction to responsive web design principles, CSS media queries and viewport settings, Flexbox and CSS Grid for flexible layouts, Introduction to Bootstrap or other CSS frameworks

**Unit II: JavaScript and Web Optimization 12 Lecture Hours**

Introduction to JavaScript: variables, data types, control flow, Functions and scope, DOM manipulation and event handling, Introduction to jQuery or other JavaScript libraries, Working with arrays and objects, Asynchronous programming and AJAX, Introduction to ES6+ features (arrow functions, modules, etc.), Introduction to modern JavaScript frameworks (React, Vue.js, Angular), Understanding web performance metrics, Techniques for optimizing CSS and JavaScript, Asset optimization (images, fonts, etc.), Introduction to caching and CDNs

**Unit III: Frontend: Building and Testing 12 Lecture Hours**

Introduction to task runners (Gulp, Grunt) or bundlers (Webpack, Parcel), CSS preprocessors (Sass, Less) for enhanced styling, Introduction to version control systems (Git), Accessibility and Usability, Principles of web accessibility, Techniques for creating accessible web content, Usability best practices and user-centered design principles, Introduction to frontend testing frameworks (Jest, Mocha), Unit testing and integration testing, Testing user interfaces and interactions.

**Unit IV: JSX and Redux 12 Lecture Hours**

Why JSX, Embedding JavaScript, Expression in JSX, JSX as an Expression, Nested elements in JSX, JSX, Attributes, JSX Comments, JSX Styling and representation as object, The State of the Component, Defining State, Changing the State, Props, Validation, Validators, Elements, Rendering Element, About render (), Creating React Element, Updating Element, components, Introducing Components, Types of Components, Functional Component, Functional Components as Stateless, Using Functional Component, Redux Concepts, Redux Principles, Data Flow, Actions, Functions, Reduces, Testing , Dev-Tools, React & Redux Integrate.

**Unit V: Web Security 12 Lecture Hours**

Introduction to web security principles, SPA frameworks, authentication and authorization systems, API analysis, detecting frameworks and libraries, Common Security Issues (eg. Cross-Site Scripting, CSRF, XXE, Injection), Common Countermeasures (e.g., Authentication, Authorization, HTTPS), securing web applications, reviewing code for security, vulnerability discovery and management, Web Application Firewalls.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Stefan Baumgartner, “Front End Tooling with Gulp”, Manning Publication. 2. “HTML5 Black Book”, Dreamtech Publications, 2016. 3. Ben Frain, “Responsive Web Design with HTML5 and CSS”, 4th Edition, Packt Publication, 2022. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Frontend Development Lab**

**List of Experiments**

**Experiment-1**: Write a program to create a simple webpage using HTML.

**Experiment-2:**  Write a program to create a website using HTML CSS and JavaScript

**Experiment-3:**  Write a program to build a Chat module using HTML CSS and JavaScript

**Experiment-4**: Write a program to create a simple calculator Application using React JS

**Experiment-5**: Write a program to create a voting application using React JS

**Experiment-6**: Write a program to create and Build a Password Strength Check using Jquery

**Experiment-7**: Write a program to create and Build a star rating system using Jquery

**Experiment-8**: Create a Simple Login form using React JS

**Experiment-9**: Using the CMS users must be able to design a web page using the drag and drop method

**Experiment-10**: Create a project on Grocery delivery application

**Experiment-11**: Connecting our TODO React js Project with Firebase

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Stefan Baumgartner, “Front End Tooling with Gulp”, Manning Publication, 2016. 2. ‎DT Editorial Services, “HTML5 Black Book”, 2nd Edition, Dreamtech Publications, 2016. 3. Ben Frain, “Responsive Web Design with HTML5 and CSS”, 4th Edition, Packt Publication, 2022. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSFS3008P** | **Backend Development** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Frontend Devlopment - CSFS2004** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the fundamentals of web development and back-end technologies.

2. Understand and explore backend frameworks, databases and data modelling.

3. Design and Implement APIs and API related tasks.

4. Utilize back-end frameworks and libraries to streamline development processes.

5. Implement best practices for security, testing and debugging.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Understand Back End Development Fundamentals

**CO2:** Create and build web pages and applications.

**CO3:** Utilize frameworks and APIs.

**CO4:** Apply security best practices and conduct testing.

**CO5:** Demonstrate effective team collaboration skills.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | **-** | **-** | 0.4 | **-** | 2 | **-** | **-** | **-** | 0.6 | **-** | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Backend Development, 12 Lecture Hours**

**Server-Side Programming**

Overview of backend development and its importance, Client-server architecture and communication protocols, Introduction to HTTP and RESTful APIs, Introduction to backend languages and frameworks (e.g., Python, Node.js, Django, Flask), Introduction to server-side programming languages (e.g., Python, JavaScript), Basics of handling HTTP requests and responses, Managing server-side sessions and cookies, Introduction to server-side rendering and template engines.

**Unit II: Data Management, API Development 12 Lecture Hours**

**and Frameworks**

Introduction to databases (relational and NoSQL), Designing database schemas and models, Performing CRUD (Create, Read, Update, Delete) operations on databases, Query optimization and indexing, designing, and implementing RESTful APIs, Authentication and authorization mechanisms, Handling API requests and responses, API versioning and documentation.

**Unit III: Frameworks and Performance Optimization** **12 Lecture Hours**

Introduction to popular backend frameworks (e.g., Express.js, Django, Ruby on Rails), Working with middleware and routing, integrating third-party APIs and services, Testing and debugging backend applications, understanding security vulnerabilities and best practices, Implementing secure authentication and authorization mechanisms, Caching and performance optimization techniques, Error handling and logging.

**Unit IV: Deployment and DevOps 12 Lecture Hours**

Overview of deployment environments (e.g., local, cloud, containers), Monitoring and scaling backend applications, Basics of containerization (e.g., Docker), Continuous integration and deployment (CI/CD), Managing infrastructure configurations, setting up CI/CD pipelines, Building and packaging applications, Automated testing and code quality checks, Continuous Deployment and Release Management, Strategies for releasing software updates, Managing deployment environments, Release orchestration and rollback strategies.

**Unit V: Advanced Topics 12 Lecture Hours**

Real-time communication with Web-Sockets, Microservices architecture and design patterns: Overview of microservices architecture and its benefits, Service discovery and service registration, Implementing communication between microservices (REST, messaging), Serverless computing: Understanding serverless architecture and its advantages, Building serverless backend systems using platforms like AWS Lambda or Google Cloud Functions, Event-driven serverless architecture and integration with other services, Function-as-a-Service (FaaS), Performance profiling and optimization strategies.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. David Herron, “Node.js Web Developent” 5th Edition, Packt Publication, 2020. 2. DT Editorial Services, “HTML5 Black Book”, 2nd Edition, Dreamtech Publications, 2016. 3. Sam Newman, “Building Microservices”, 2nd Edition, O’Reilly Publication, 2021. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Backend Development Lab**

**List of Experiments**

1. Create a web page with all possible elements of HTML5
2. Create a web page with all types of Cascading style sheets
3. Create a Responsive Web page with HTML and CSS
4. Create Responsive web page with Bootstrap
5. Programs to demonstrate JavaScript array, object and functions
6. Client-Side Scripts for Form Validation using JavaScript
7. Programs to familiarise ES6 concepts
8. Programs to demonstrate DOM and event handling.
9. Programs using AJAX with HTML, XML and JSON data
10. Programs to familiarize Query.(2 program)
11. Create a website with HTML, CSS and JavaScript (implement Ajax)
12. Programs to familiarize Server-Side Scripting using Node JS
13. Programs using Mongo DB database with Node JS

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. David Herron, “Node.js Web Developent”, 5th Edition, Packt Publication, 2020. 2. DT Editorial Services “HTML5 Black Book”, 2nd Edition, Dreamtech Publications, 2016. 3. Sam Newman, “Building Microservices”, 2nd Edition, O’Reilly Publication, 2021. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSFS3009P** | **Microservices and Spring-Boot** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | **Backend development - CSFS3008P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Develop Microservices with Spring Boot

2. Implement Service Discovery and Load Balancing

3. Implement Event-Driven Communication and develop API gateway

4. Monitor, Test and Deploy microservices effectively

5. Develop Troubleshooting and Debugging abilities.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Understanding service-oriented architecture and its utility

**CO2**. Compare microservices with SOA

**CO3.** Design patterns of architecture for a microservices led approach for scaling IT solutions.

**CO4**. Establish role of CI/CD with containers

**CO5:** Demonstrate project development skills.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | **-** | **-** | 0.4 | **-** | 2 | **-** | **-** | **-** | 0.6 | **-** | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Microservices & Spring Boot 10 Lecture Hours**

Spring Boot Introduction, Introduction, Software Architectures, Microservices, Architecture Layers, Application Layer, Business Layer, Enterprise Layer, Need of SpringBoot and its stakeholders, Understand architectural patterns and styles, Explain monolithic architecture, strengths and limitations, Describe SOA architecture, strengths and limitations, Discuss SOA components and its principles, Identify microservices architecture, strengths and limitations,

**Unit II: Microservices Components 10 Lecture Hours**

List microservices components and its principles, define domain driven design concepts, explain strategic and tactical design, discuss domain, bounded context, ubiquitous language and context mapping, Describe the building blocks of DDD, its strengths and limitations, Explain how domain driven design applies to microservices, Discuss designing services applying DDD concepts, Define service communication, Describe the API gateway

**Unit III: Spring Framework 10 Lecture Hours**

Advantages of Spring Framework, Spring Modules, IoC Containers, Bean Factory, Spring Boot, Advantages of Spring Boot, Create Projects in Spring Boot, Angular Components, Angular Forms, Services & Dependency, Modules, Advanced Components, Handling Errors, SEO and Angular, Angular CLI

**Unit IV: REST and Spring Boot 10 Lecture Hours**

Web Services, Terminology, SOAP, Restful Web Service, SOAP Vs RESTful Web Services, RESTful Web Services with SpringBoot, User Bean and User Service, GET Method, POST Method, Validations, Initialize REST API, Connect Angular Front with RESTful API, HTTP Services, Retrieval, Spring Security, Authentication Service, JWt Framework, JPA and Hibernate

**Unit V: Implementing Microservices and Case Studies 5 Lecture Hours**

Best practices for designing and implementing microservices, single responsibility principle, asynchronous communication: loose coupling, fault tolerance, backward compatibility, organizational efficiencies, case studies of real-world microservices architectures (Netflix; Amazon; Uber), future trends and advancements in microservices.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Sam Newman, “Building Microservices: Designing Fine-Grained Systems”, 2nd Edition, O’Reilly Publication, 2021. 2. K. Siva Prasad Reddy, “Spring Boot: Applications and Microservices”, APress Publication. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Microservices and Spring-Boot Lab**

**List of Experiments**

**Experiment 1**: Setting Up the Development Environment (Java, Spring Boot, IDE, Docker, etc.)

**Experiment 2**: Creating a Basic Spring Boot Application

**Experiment 3**: Building a RESTful API

**Experiment 4:** Implementing Service Discovery with Eureka

**Experiment 5:** Load Balancing with Ribbon

**Experiment 6:** Containerizing Microservices with Docker

**Experiment 7:** Implementing Event-Driven Communication with Spring Cloud Stream

**Experiment 8:** Setting Up Centralized Configuration with Spring Cloud Config

**Experiment 9:** Implementing API Gateway with Spring Cloud Gateway

**Experiment 10:** Monitoring Microservices with Spring Boot Actuator

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. K Siva Prasad Reddy and Springerlink (Online Service, Beginning Spring Boot 2 : Applications and Microservices with the Spring Framework. Berkeley, Ca: Apress, 2017. |
| **Reference books** | 1. S. Newman, Building Microservices. Beijing: O’reilly, 2015. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSVT4023P** | **Cloud Computing and Security** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Cryptography and Network Security - CSEG3040** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

This course will provide a foundational understanding of what is required to secure a cloud ecosystem, regardless of the vendor. The concepts and principles discussed will help bridge the gaps between traditional and cloud architectures while accounting for the shifting thought patterns involving enterprise risk management. Students who complete this course will enter into any organization utilizing the cloud and immediately bring value to the infrastructure and security teams.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Understand core cloud computing concepts and fundamental principles, including standard delivery models and service designs.

**CO2:** Identity and access management practices of both cloud providers and consumers.

**CO3:** Design the foundational security practices that are required to secure modern cloud computing infrastructures.

**CO4:** Implement regulatory requirements needed to secure data in the cloud and the difficulties in meeting those requirements.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 2 | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | 3 | **-** | **-** | 2 | **-** | 3 |  |  | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | 2 |  | **-** | 1 | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | 1 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | 0.6 | 1.4 | 0.6 | **-** | 2 | **-** | 0.6 |  | 0.6 | 0.4 | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Cloud Computing Overview 12 Lecture Hours**

Origin & History of Cloud Computing, Cloud & Cloud Engineering, Evolution, Characteristics of Cloud, Cloud Computing Elements, Types of Cloud Computing, Trends in Cloud Computing, Cloud Service Providers, Traditional IT Providers, Cloud & DevOps, Benefits of Cloud Computing, Properties, Disadvantages, Amazon Web Services( AWS)

**Unit II: Cloud Computing Architecture 12 Lecture Hours**

Traditional Vs Cloud Computing Architecture, How Cloud Computing Works, Networking in Cloud, Deployment Models, IaaS (Infrastructure as a Service) & its Resource Virtualization examples, PaaS (Platform as a Service), SaaS (Software as a service), SaaS (Storage as a Service), difference between IaaS, PaaS, & SaaS, Cloud Platform & Management, Service Management in Cloud Computing, Service Level Agreements (SLA’s) , Scaling support in cloud / elastic nature, setting budget & notifications for budgets, Global Infrastructure of any cloud in general (Regions, Availability Zones, Data Centres)

**Unit III: AWS Basic Architecture 12 Lecture Hours**

Elastic Compute Cloud (EC2), Elastic Block Storage (EBS), Elastic Load Balancing (ELB), Security Groups, Elastic Cache, Amazon RDS, Storage & Backups, Scaling, Amazon Virtual Private Cloud (VPC), Features of VPC, Subnets, Content Delivery Network (CDN using Cloud Front), Features, Cloud Watch, Beanstalk, route 53, S3, Auto Scaling Groups

**Unit IV: Cloud Security 12 Lecture Hours**

Security Concepts, Security Planning, Boundaries, Identity and Access Management(IAM), Cloud Security essentials, Cloud Security Alliance, Data Security, Encryption, Isolated Access to Data, Introduction to Key Management Service (KMS) of AWS

**Unit V: Role of Cloud in DevOps 12 Lecture Hours**

The Role of Cloud in DevOps, Cloud for successful Ops, Secure Cloud Platforms for DevOps, Cloud & IT Budgets, building a business case for cloud computing, Infrastructure as Code (IAC)

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Tim Mather, Subra Kumaraswamy, and Shahed Latif, “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly Media, 2019. 2. Rajkumar Buyya, James Broberg, and Andrzej Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley, 2013. 3. Dan C. Marinescu, “Cloud Computing: Theory and Practice”, Morgan Kaufmann, 2013. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Cloud Computing and Security Lab**

**List of Experiments**

**Experiment 1:** Building your first Virtual Private Cloud-AWS

**Experiment 2:** Delegate access across AWS accounts using roles-AWS

**Experiment 3:** Create a customer managed policy-AWS

**Experiment 4:** Use attribute-based access control (ABAC)-AWS

**Experiment 5:** Permit users to manage their credentials and MFA settings (AWS)

**Experiment 6**: Implement Azure key vault

**Experiment 7:** Manage Access to Azure with Role-Based Access Control

**Experiment 8:** Manage Identity and Access- MFA, and Conditional Access (Azure)

**Experiment 9:** Manage Identity and Access- Implement Directory Synchronization (Azure)

**Experiment 10:** Smart Access Control on GCP: Web, programmatic, and command-line access

**Experiment 11:** Write an IAM policy by using client libraries (GCP)

**Experiment 12:** Using AWS WAF to control access to your content (AWS)

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. R. L. Krutz, Russell Dean Vines, and J. Wiley, Cloud security : a comprehensive guide to secure cloud computing. New Delhi: Wiley, 2016. 2. T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy : an Enterprise Perspective on Risks and Compliance. Sebastopol: O’Reilly Media, Inc., 2009. |
| **Reference books** | 1. I. M. Abbadi, Cloud Management and Security. John Wiley & Sons, 2014. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDV4015P** | **Container Orchestration and Security** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | Microservices and Spring-Boot - CSFS3009P | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The Container Orchestration with Kubernetes and Security course will help the students to grasp the key skills, technology, and concepts that a Kubernetes administrator needs to know. Plan to oversee containerized workloads and administrations with industrial Organizations utilizing the Kubernetes Preparing course. It covers all the aspects, including application lifecycle management, installation, configuration and validation, networking, scheduling, security, cluster maintenance, core concepts, Azure Kubernetes, storage, and more.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1:** Understand the significance of containers over virtual machines.

**CO2:** Design the container-based deployment of an application on container-based platform.

**CO3:** Implement a Kubernetes Cluster and deploy a Dockerized application on a cloud based Kubernetes

cluster.

**CO4:** Test and Integrate security protocols in containersed application for any vulnerability.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 3 | 2 | **-** | **-** | 3 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 2** | **-** | 3 | **-** | **-** | 2 | **-** | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO 3** | **-** | 2 | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** |
| **CO4** | **-** | **-** | 1 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **-** | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | 2 | **-** | **-** | **-** |
| **Average** | 0.6 | 1.4 | 0.6 | **-** | 2 | **-** | 0.6 | **-** | 0.6 | 0.4 | **-** | 0.4 | **-** | **-** | **-** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Application Containerization 12 Lecture Hours**

Understanding Containers, Challenges in the Software Industry, Problems in Software Industry Before Containers, Put that in Container! Solution by containers in the Software Industry, Virtualization, Hypervisor, Scope of Virtualization, Containers vs Virtual Machines, Understanding Containers, Containerization Platform, Runtime and Images, Container Platform, Container Runtime, The Chroot System, FreeBSD Jails, Linux Containers (LXC), Docker

**Unit II: Introduction to Containerization and different environments 12 Lecture Hours**

Docker architecture, Docker Daemon (Container Platform), Docker Rest API, Understanding Different environments: (Dev, QA and Prod), Overcoming issues with different environments, Development Environment, Testing Environment, Staging Environment, Production Environment, Virtual machines for dev/deployments, Containers for dev/deployments, Advantages and drawbacks of containerization

**Unit III: Docker Fundamentals and Internals 12 Lecture Hours**

Docker container states, docker image vs docker containers, docker image creation using docker commit & Dockerfile, Dockerfile important keywords, docker tags, persistent storage use-case, docker volumes, docker networks, creating custom networks in docker, docker registry, docker inbuilt security concepts (namespaces, cgroups)

**Unit IV: Orchestration Tools 12 Lecture Hours**

What is orchestration, Need of orchestration, Case study: Need of Orchestration, Need of Orchestration: Container and Microservices, Docker Swarm and Kubernetes, Architecture, AWS (ECS,EKS), AWS Elastic Container Services Architecture, Azure Kubernetes Services, OpenShift, KUBERNETES ON CLOUD, Monitoring of container, How to monitor

**Unit V: Container Security 12 Lecture Hours**

Docker vs Vagrant, Docker Challenges Revisited, Vulnerabilities in images (Public and Private), Denial of service attacks, Privilege escalation methods in Docker, Security misconfigurations, Container Security, Content Trust and Integrity checks, Capabilities and namespaces in Docker, Segregating Networks, Kernel Hardening using SecComp and AppArmor, Static Analysis of container (Docker) images, Dynamic Analysis of container hosts and daemons.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | * + - 1. 1 Liz Rice, “Container Security Fundamental Technology Concepts that Protect Containerized Applications”, O'Reilly Media, 2020.       2. [José Manuel Ortega Candel](https://www.google.co.in/search?hl=en&sxsrf=AB5stBhwOJSswlEbX0Ftl6ke0VJO6wMC8Q:1688746781851&q=inauthor:%22Jos%C3%A9+Manuel+Ortega+Candel%22&tbm=bks), “DevOps and Containers Security and Monitoring in Docker Containers”, BPB Publication, 2020.       3. Gigi Sayfan, “Mastering Kubernetes”, 3rd Edition, [Packt Publishing](https://www.google.co.in/search?hl=en&gbpv=1&dq=Kubernetes+Container+Orchestration&printsec=frontcover&q=inpublisher:%22Packt+Publishing%22&tbm=bks&sa=X&ved=2ahUKEwim1oiegP3_AhVV8zgGHftbBg8QmxMoAHoECCUQAg&sxsrf=AB5stBgLYxiSvNSj3eIURHr0R8VoTdBNaw:1688746844435), 2020. |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Container Orchestration and Security Lab**

**List of Experiments**

1. Installing Vagrant & Creating basic vagrant box using VirtualBox virtualization.
2. Understanding vagrant file - Configuration - CPU, RAM, Storage, Provisioning (Shell Script).
3. Docker Machine - Installation , configuration, creating machines (on VirtualBox).
4. Docker - Installation, Configuration, Running Images.
5. Dockerfile - Containerizing application, Building Images, Tagging, Publishing.
6. Docker - Volumes, Env, Monitoring (Docker stats).
7. DTR - Docker Hub, Private Registries, Publishing images.
8. Docker Compose - Installation, Creating Compose files, Running Images using docker-compose.
9. Running Multi-Container applications using docker compose and on Swarm.
10. Kubernetes -Minikube installation and fundamentals.
11. Deploying Pods and Services on minikube.
12. Build Docker Image using .Dockerignore file.
13. Prepare and Implement Docker Container Restart Policy
14. Working with Metadata, Log File using Docker

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Application containerization by Xebia |
| **Reference books** | . Jarosław Krochmalski, “Developing with Docker”, Packt Publishing, 2016.  2. Adrian Mouat, “Orchestrating, clustering, and managing containers”, O'Reilly Media, Inc., 2016 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % |  |

1. **Cyber Security and Forensics Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSSF2015P** | **Information Technology and Cyber Security** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

This course on Information Technology and Cyber Security aims to provide students with a comprehensive understanding of fundamental concepts and principles in information security, types of cybercrimes, IT security planning, audit, and compliance, network security and data privacy, as well as physical security. The course objectives include developing knowledge and skills in risk management, security controls and technologies, incident response, legal and ethical considerations, and emerging trends in information security. Students will also gain insights into cybersecurity laws, network attacks, data encryption, secure transmission protocols, and physical security measures. Practical exercises and real-world examples will enhance critical thinking, problem-solving, and communication skills, fostering professionalism and ethical responsibility in the field.

**Course Outcomes**

On completion of this course, the students will be able to

1. Understand the fundamental concepts and principles of information security, including risk management, security controls, and incident response.
2. Identify and analyze different types of cyber-crimes, including hacking, malware attacks, identity theft, and financial cyber-crimes, and comprehend the impact of these crimes on individuals and organizations.
3. Apply IT security planning, audit, and compliance practices, including security policies and standards, risk assessment, and incident response, to ensure effective management and protection of information technology resources.
4. Evaluate and implement network security measures, data privacy principles, and secure transmission protocols to safeguard networks, web applications, and cloud computing environments.
5. Recognize the importance of physical security and employ appropriate measures, such as access control systems, video surveillance, and emergency response procedures, to mitigate physical security threats and vulnerabilities.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |  | 3 | **-** |
| **CO 3** | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 |  | **-** |
| **CO4** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO5** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 |
| **Average** | **-** | **-** | **-** | 0.4 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 0.4 | 1 | 0.8 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Overview & Fundamentals of Information Security 12 Lecture Hours**

Introduction to Information Security, Information Security Principles, Security Threats and Vulnerabilities, Risk Management, Security Policies and Procedures, Security Controls and Technologies, Security Incident Response, Legal and Ethical Considerations, Emerging Trends in Information Security.

**Unit II: Types of Cyber Crimes 12 Lecture Hours**

Introduction to Cyber Crimes, Hacking and Unauthorized Access, Malware Attacks, Identity Theft and Fraud, Data Breaches and Information Leakage, Online Harassment and Cyberbullying, Financial Cyber Crimes, Cyber Terrorism and State-sponsored Attacks, Cybersecurity Laws and International Cooperation.

**Unit III: IT Security Planning, Audit & Compliance 12 Lecture Hours**

Introduction to IT Security Planning, Audit & Compliance, IT Security Planning, IT Security Policies and Standards, IT Security Governance, IT Security Audits, Compliance Management, Security Risk Assessment and Management, Security Incident Response and Forensics, Security Awareness and Training Programs, Security Metrics and Reporting

**Unit IV: Introduction to Network Security & Data Privacy 12 Lecture Hours**

Introduction to Network Security & Data Privacy, Network Security Fundamentals, Types of Network Attacks, Network Security Devices and Technologies, Firewalls and Intrusion Detection Systems, Virtual Private Networks (VPNs) and Secure Remote Access, Wireless Network Security, Web Application Security, Data Privacy Principles and Regulations, Data Encryption and Cryptography, Secure Data Transmission Protocols, Data Backup and Disaster Recovery, Privacy Enhancing Technologies, Privacy Policies and Consent Management, Securing Cloud Computing Environments.

**Unit V: Introduction to Physical Security 12 Lecture Hours**

Introduction to Physical Security, Importance of Physical Security, Physical Security Threats and Vulnerabilities, Risk Assessment and Site Survey, Access Control Systems, Video Surveillance and CCTV Systems, Intrusion Detection and Alarm Systems, Perimeter Security and Fencing, Security Lighting and Signage, Visitor Management and Access Procedures, Locks and Keys, Security Personnel and Guard Services, Emergency Response and Crisis Management, Security Policies and Procedures, Physical Security Audits and Inspections.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Duane C. Wilson, “Cybersecurity” The MIT Press, 2021. 2. David Alexander, Amanda Finch, David Sutton, and Andy Taylor, “Information Security Management Principles”, 2nd Edition, BCS, The Chartered Institute for IT, 2013. |
| **Reference books** | 1. Dr. Erdal Ozkaya, “Cybersecurity: The Beginner's Guide: A comprehensive guide to getting started in cybersecurity”, Packt publisher, 2019. 2. Charles J. Brooks, Christopher Grow, Philip Craig, and Donald Short, “Cybersecurity Essentials”, Sybex, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Information Technology and Cyber Security Lab**

**List of Experiments**

**Experiment** 1. System Event Logs

**Experiment** 2. Trace Email Sender Location

**Experiment** 3. Live Vulnerability on Internet devices

**Experiment** 4. Basic Footprinting

**Experiment** 5. Hack using Search Engines

**Experiment** 6. Create Irritating Batch Files

**Experiment** 7. Local Vulnerability Scanning

**Experiment** 8. Kali Linux

**Experiment** 9. Network & Host Scanning

**Experiment** 10. Vulnerability Scanning using Nessus

**Experiment** 11. Phishing Email Analysis.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSSF3030P** | **Ethical Hacking & Penetration Testing** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Information Technology and Cyber Security -** CSSF2015P | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The objective of the course is to provide students with a comprehensive understanding of the principles and techniques involved in ethical hacking and penetration testing. The course aims to equip students with the knowledge and skills necessary to identify vulnerabilities, exploit weaknesses, and assess the security of computer systems, networks, and applications. Students will learn about various hacking methodologies, tools, and technologies used in ethical hacking and penetration testing. The course will emphasize hands-on practical exercises, simulations, and real-world scenarios to develop the ability to assess, analyze, and secure systems from potential threats and vulnerabilities. Additionally, students will gain insights into legal and ethical considerations in conducting ethical hacking activities and will be prepared to apply their knowledge to protect and secure organizations' digital assets.

**Course Outcomes**

On completion of this course, the students will be able to

1. Understand the principles, methodologies, and techniques of ethical hacking and penetration testing to identify and exploit vulnerabilities in computer systems, networks, and applications.
2. Demonstrate proficiency in using various tools, technologies, and frameworks commonly employed in ethical hacking and penetration testing, including reconnaissance, scanning, enumeration, and exploitation.
3. Develop critical thinking and problem-solving skills to effectively analyze and assess security vulnerabilities and recommend appropriate countermeasures to enhance system security.
4. Apply ethical and legal considerations in conducting ethical hacking activities, adhering to professional codes of conduct, and protecting sensitive information during penetration testing engagements.
5. Demonstrate effective communication skills in documenting and presenting findings, risks, and recommendations resulting from ethical hacking and penetration testing activities to stakeholders in a clear and concise manner.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 3 |
| **CO 2** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 1 |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** |
| **CO4** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** |
| **CO5** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** |
| **Average** | **-** | **-** | **-** | 0.4 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 0.4 | 1.4 | 0.8 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Ethical Hacking 10 Lecture Hours**

Introduction to Ethical Hacking, Ethical Hacking vs. Unethical Hacking, Legal and Ethical Considerations, Phases of Ethical Hacking, Footprinting and Reconnaissance, Scanning and Enumeration, Vulnerability Assessment and Penetration Testing, System Hacking and Exploitation, Web Application Security Testing, Wireless Network Security Testing, Social Engineering Techniques, Physical Security Assessments, Security Tools, and Technologies, Reporting and Remediation.

**Unit II: Setting up Ethical Hacking Labs & Targets 12 Lecture Hours**

Setting up Ethical Hacking Labs & Targets, Understanding Lab Requirements, Virtualization Technologies for Hacking Labs, Building a Virtual Lab Environment, Configuring Network Segmentation, Creating Target Machines and Vulnerable Systems, Lab Network Design Considerations, Lab Setup for Web Application Security Testing, Lab Setup for Wireless Network Security Testing, Lab Setup for System and Network Exploitation, Lab Setup for Social Engineering Exercises, Lab Maintenance and Security Best Practices, Lab Documentation and Reporting.

**Unit III: Vulnerability Assessment & Pen Testing Basics**  **14 Lecture Hours**

Vulnerability Assessment & Pen Testing Basics, Introduction to Vulnerability Assessment, Importance and Benefits of Penetration Testing, Types of Vulnerability Assessments, Scoping and Planning a Penetration Test, Pre-engagement Activities, Information Gathering and Reconnaissance, Vulnerability Scanning and Assessment, Exploitation and Post-Exploitation Techniques, Web Application Penetration Testing, Network Penetration Testing, Wireless Network Penetration Testing, Social Engineering Testing, Report Generation and Recommendations.

**Unit IV: Overview of various VA/PT Tools 12 Lecture Hours**

Overview of various VA/PT Tools, Nessus, OpenVAS, NMAP, Qualys, Burp Suite, Acunetix, Metasploit, Wireshark, OWASP ZAP, Nikto, SQLMap, DirBuster, BeEF, Hydra, John the Ripper, Aircrack-ng, THC-Hydra, Hashcat, Armitage.

**Unit V: Unit-5 Scanning, Enumeration, Attacking 12 Lecture Hours**

Scanning, Enumeration, Attacking, Port Scanning Techniques, Network and Host Discovery, Vulnerability Scanning, Service Enumeration, Banner Grabbing, Operating System Fingerprinting, Web Application Scanning, DNS Enumeration, SMTP Enumeration, SMB Enumeration, SNMP Enumeration, LDAP Enumeration, Exploitation Techniques, Brute-Force Attacks, Password Cracking, Remote Code Execution, Privilege Escalation, Denial of Service Attacks.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Jon Erickson, “Hacking: The Art of Exploitation”, 2nd Edition, No Starch Press, US, 2008 2. Patrick Engebretson, “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy”, 2nd Edition, Syngress, 2013. |
| **Reference books** | 1. Rafay Baloch, “Ethical Hacking and Penetration Testing Guide”, Auerbach Publications, 2014. 2. Keshav Kaushik, and Akashdeep Bhardwaj, “Perspectives on Ethical Hacking and Penetration Testing”, IGI Global, 2023. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Ethical Hacking & Penetration Testing Lab**

**List of Experiments**

1. Footprinting Basics
2. Footprinting using SHODAN
3. Footprinting Tools
4. Advanced Footprinting
5. Google Hacking
6. Scanning using NMAP
7. Vulnerability Assessment
8. EHPT Lab Setup
9. DVWA Setup
10. Exploit DVWA

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Network Security Practices** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 08** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Data communication and Networks - CSEG2065** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

This course aims to provide students with a comprehensive understanding of the principles, technologies, and best practices involved in securing computer networks. The course objectives are to equip students with the necessary knowledge and skills to identify and mitigate network security threats, implement appropriate network security controls and technologies, and respond effectively to network security incidents. Students will learn about the importance and challenges of network security, the goals and objectives of network security, and the common threats and vulnerabilities faced by networks. They will explore network protocols, architecture, and models, gaining an understanding of network devices, components, and segmentation techniques. The course will cover network security technologies and tools such as firewalls, intrusion detection/prevention systems (IDS/IPS), virtual private networks (VPNs), and network access control (NAC). Students will also learn about network access control and authentication methods, wireless network security, and guest access management. Additionally, the course will cover topics like firewall and IDS/IPS systems, VPNs and secure remote access, network security best practices, incident response, and forensics in network security. By the end of the course, students will be proficient in designing, implementing, and maintaining secure network environments, effectively mitigating network security risks, and responding to network security incidents in a timely and efficient manner.

**Course Outcomes**

On completion of this course, the students will be able to

1. Understand the fundamental concepts and principles of network security, including the importance and challenges of securing computer networks, as well as the goals and objectives of network security.
2. Identify and assess common network threats and vulnerabilities, including malware and virus attacks, denial of service (DoS) attacks, network intrusions, and unauthorized access, and implement appropriate measures to mitigate these risks.
3. Demonstrate proficiency in deploying and configuring network security technologies and tools, such as firewalls, intrusion detection/prevention systems (IDS/IPS), virtual private networks (VPNs), and network access control (NAC), to protect network infrastructure and ensure secure network communications.
4. Apply authentication methods and access control models to manage network user authentication and authorization, including password-based, biometric, and token-based authentication, as well as role-based access control (RBAC), mandatory access control (MAC), and discretionary access control (DAC).
5. Develop the knowledge and skills to implement network security best practices, including the establishment of security policies and procedures, network hardening techniques, security auditing, compliance management, and incident response procedures. Students will also gain an understanding of network forensics and its role in investigating network security incidents.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 |  | 3 |
| **CO 2** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |  | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO4** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** |
| **CO5** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** |
| **Average** | **-** | **-** | **-** | 0.4 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 0.6 | 1.2 | 1 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Network Security 6 Lecture Hours**

Overview of Network Security, Importance and Challenges of Network Security, Goals and Objectives of Network Security, Network Security Threat Landscape, Network Security Controls and Technologies.

**Unit II: Network Protocols and Architecture 8 Lecture Hours**

Introduction to Network Protocols, TCP/IP Protocol Suite, Network Architecture and Models (OSI, TCP/IP), Network Devices and Components, Network Segmentation and Zones.

**Unit III: Network Threats and Vulnerabilities 8 Lecture Hours**

Types of Network Threats, Common Network Vulnerabilities, Malware and Virus Attacks, Denial of Service (DoS) Attacks, Network Intrusions and Unauthorized Access.

**Unit IV: Network Security Technologies and Tools 8 Lecture Hours**

Firewalls and Intrusion Detection/Prevention Systems (IDS/IPS), Virtual Private Networks (VPNs), Network Access Control (NAC), Secure Sockets Layer/Transport Layer Security (SSL/TLS), Network Security Appliances.

**Unit V: Network Access Control and Authentication 8 Lecture Hours**

Authentication Methods (Passwords, Biometrics, Tokens), Access Control Models (RBAC, MAC, DAC), Network Access Policies, Wireless Network Security, Guest Access Management.

**Unit VI: Firewall and Intrusion Detection/Prevention Systems 8 Lecture Hours**

Introduction to Firewalls, Firewall Types (Packet Filtering, Stateful Inspection, Application-Level), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), Firewall Rule Management.

**Unit VII: Virtual Private Networks (VPNs) and Secure 8 Lecture Hours**

**Remote Access**

VPN Fundamentals, VPN Protocols (IPSec, SSL/TLS), Site-to-Site VPNs, Remote Access VPNs, Secure Remote Access Technologies (RDP, SSH), VPN Configuration and Troubleshooting.

**Unit VIII: Network Security Best Practices and 6 Lecture Hours**

**Incident Response**

Security Policies and Procedures, Network Hardening Techniques, Security Auditing and Compliance, Incident Response and Handling, Forensics in Network Security.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. William Stallings, “Cryptography and Network Security - Principles and Practice”, 7th Edition, Pearson, 2017. 2. William Stallings, “Network Security Essentials”, 6th Edition, Pearson, 2018. |
| **Reference books** | 1. E Cole, “Network Security Bible”, 2nd Edition, John Wiley & Sons Inc, 2009. 2. Atul Kahate, “Cryptography And Network Security”, 4th Edition, McGraw-Hill, 2019. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Network Security Practices Lab**

**List of Experiments**

1. Troubleshoot Network without Tools
2. Network Packet Sniffing, Capture & Analysis
3. Trace network queries
4. HTTP Traffic tracing
5. Decrypt SSL Traffic
6. Crack Passwords of ZIP, RAR, PDF, Shadow files
7. TCP Analysis
8. ICMP & FTP Analysis
9. UDP Analysis
10. DHCP & DNS Analysis

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. William Stallings, “Cryptography and Network Security - Principles and Practice”, 7th Edition, Pearson, 2017. 2. William Stallings, “Network Security Essentials”, 6th Edition, Pearson, 2018. |
| **Reference books** | 1. E Cole, “Network Security Bible”, 2nd Edition, John Wiley & Sons Inc, 2009. 2. Atul Kahate, “Cryptography And Network Security”, 4th Edition, McGraw-Hill, 2019. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSSF4021P** | **Digital Forensics** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 10** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Ethical Hacking & Penetration Testing-**  **CSSF3030P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

This course aims to provide students with a comprehensive understanding of the principles, methodologies, and techniques involved in digital forensics investigations. The course objectives are to equip students with the necessary knowledge and skills to conduct digital forensic examinations, preserve and collect digital evidence, analyze various digital artifacts, and prepare clear and concise forensic reports. Students will learn about the fundamental concepts of digital forensics, including the forensic process, evidence preservation, and acquisition techniques. They will gain expertise in analyzing file systems, operating systems, network traffic, mobile devices, and memory. The course will cover essential technical concepts such as disk imaging, hashing, metadata analysis, volatile memory forensics, network protocols, and encryption. Students will also explore the requirements for setting up a computer forensics lab, including hardware, software, storage, and lab security. Legal and ethical considerations in digital forensics, anti-forensics techniques, and report writing skills will be emphasized throughout the course. By the end of the course, students will be proficient in conducting digital investigations, handling digital evidence, and producing professional forensic reports in compliance with legal and ethical standards.

**Course Outcomes**

On completion of this course, the students will be able to

1. Understand the fundamental concepts and principles of digital forensics, including the forensic process, evidence preservation, and legal and ethical considerations, enabling students to conduct digital investigations in a professional and responsible manner.
2. Gain proficiency in digital evidence collection, preservation, and analysis techniques, including acquiring and imaging digital devices, analyzing file systems, operating systems, network traffic, mobile devices, and memory.
3. Develop skills in utilizing forensic tools and software to analyze digital artifacts, including file metadata, email communications, web browser activities, and social media interactions, in order to extract valuable information for investigative purposes.
4. Apply essential technical concepts in digital forensics, such as disk imaging, hashing, volatile memory forensics, network protocols, and encryption, to effectively identify and interpret digital evidence, establish timelines, and reconstruct events.
5. Demonstrate the ability to produce professional forensic reports, adhering to industry standards, and effectively communicate findings and conclusions in a clear and concise manner. Students will also be aware of anti-forensics techniques and be able to detect and counteract these attempts to thwart digital forensic investigations.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 2** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | **-** | **-** |
| **CO4** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 |
| **CO5** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 |
| **Average** | **-** | **-** | **-** | 0.4 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1.2 | 0.8 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Digital Forensics 6 Lecture Hours**

Digital Forensics Fundamentals, Introduction to Digital Forensics, Digital Forensics Process and Methodology, Preservation and Collection of Digital Evidence, Acquisition and Imaging Techniques, Forensic Analysis of File Systems, Forensic Analysis of Operating Systems, Network Forensics and Traffic Analysis, Mobile Device Forensics, Memory Forensics, Malware Analysis and Reverse Engineering, Steganography and Cryptography in Digital Forensics, Incident Response and Forensics, Legal and Ethical Considerations in Digital Forensics.

**Unit II: Essential Technical Concepts 6 Lecture Hours**

Essential Technical Concepts for Digital Forensics, File Systems and Data Storage Concepts, Disk Imaging and Acquisition, File Metadata and Timestamp Analysis, File Carving and Data Recovery, Hashing and Digital Signatures, Volatile Memory Forensics, Network Forensics, Email Forensics, Mobile Device Forensics, Cloud Forensics, Steganography and Anti-Forensics Techniques, Forensic Analysis of Malware, Log Analysis and Event Reconstruction, Encryption and Decryption Techniques, Data Hiding and File Steganography, Network Protocols and Packet Analysis, Forensic Tools and Software, Evidence Handling and Chain of Custody, Legal and Ethical Considerations in Digital Forensics.

**Unit III: Hard Disks and File Systems 6 Lecture Hours**

Hard Disks and File Systems, Introduction to Hard Disks, Hard Disk Architecture and Components, Hard Disk Interfaces (IDE, SATA, SCSI, NVMe), Hard Disk Partitioning and Formatting, File System Basics, FAT File System, NTFS File System, Ext File Systems (Ext2, Ext3, Ext4), HFS+ File System, APFS File System, File System Journaling and Recovery, Disk Partitioning Schemes (MBR, GPT), Disk Imaging and Cloning, Disk Maintenance and Optimization, Disk Encryption and Data Protection, Bad Sectors and Disk Repair, File System Forensics, Recovering Deleted Files and File Metadata, File System Analysis Tools, Disk and File System Forensics Techniques, Best Practices for Hard Disk and File System Security.

**Unit IV: Requirements for a Computer Forensics Lab 6 Lecture Hours**

Requirements for a Computer Forensics Lab, Introduction to Computer Forensics Lab, Hardware Requirements for a Forensics Lab, Workstation Setup and Configuration, Storage and Backup Solutions, Network Infrastructure for a Forensics Lab, Forensic Imaging and Acquisition Tools, Forensic Analysis and Investigation Tools, Forensic Software and Applications, Virtualization for Forensic Analysis, Lab Security and Access Control, Forensic Lab Documentation and Procedures, Lab Environment and Ergonomics, Data Integrity and Chain of Custody, Lab Certification and Accreditation, Training and Skill Development, Lab Budgeting and Resource Management, Emerging Trends in Computer Forensics Labs.

**Unit V: Acquiring Digital Evidence 6 Lecture Hours**

Acquiring Digital Evidence, Introduction to Digital Evidence Acquisition, Legal Considerations and Chain of Custody, Types of Digital Evidence, Volatile and Non-Volatile Data, Acquiring Data from Hard Disks and Storage Devices, Imaging and Hashing Techniques, Live Data Acquisition, Network Traffic Capture and Analysis, Acquiring Data from Mobile Devices, Forensic Acquisition of Cloud Data, Remote Acquisition Techniques, Memory Acquisition and Analysis, Data Extraction from Embedded Systems, Data Acquisition from IoT Devices, Acquisition of Social Media and Web Content, Forensic Imaging of Virtual Machines, Challenges and Best Practices in Digital Evidence Acquisition.

**Unit VI: Analysis of Digital Evidence 6 Lecture Hours**

Analysis of Digital Evidence, Introduction to Digital Evidence Analysis, Preservation and Validation of Digital Evidence, Digital Forensic Tools and Software, File and Metadata Analysis, Email and Messaging Analysis, Web and Social Media Analysis, Network Traffic Analysis, Malware Analysis and Reverse Engineering, Memory Analysis, Mobile Device Analysis, Cloud Data Analysis, Multimedia and Image Forensics, Steganalysis, Database and Registry Analysis, Log File Analysis, Timeline Analysis, Link and Association Analysis, Data Recovery and Reconstruction Techniques, Reporting and Presentation of Digital Evidence, Challenges and Best Practices in Digital Evidence Analysis.

**Unit VII: Windows Forensic Analysis 6 Lecture Hours**Windows Forensic Analysis, Introduction to Windows Forensics, Windows File Systems (FAT, NTFS), Windows Registry Analysis, Windows Event Logs Analysis, User Account and Authentication Analysis, Windows Artifact Analysis (Recent Files, Jump Lists, Prefetch), Windows Timeline Analysis, Internet Browser Forensics (Internet Explorer, Microsoft Edge, Chrome, Firefox), Email Client Forensics (Outlook, Thunderbird), Windows Memory Analysis, Windows Malware Analysis, Windows Network Forensics, Windows Shellbags Analysis, Windows Link File Analysis, Windows Forensic Tools and Software, Anti-Forensics Techniques on Windows, Reporting and Presentation of Windows Forensic Findings.

**Unit VIII: Web Browser and E-mail Forensics 6 Lecture Hours**

Web Browser and E-mail Forensics, Introduction to Web Browser Forensics, Web Browser Architecture and Components, Web Browser Forensic Artifacts, Internet History Analysis, Bookmark and Favorites Analysis, Cookie Analysis, Downloaded Files and Cache Analysis, Form Data and Autofill Analysis, Webmail Forensics, E-mail Header Analysis, E-mail Content and Attachments Analysis, E-mail Metadata Analysis, Web Browser and E-mail Forensic Tools, Extracting Evidence from Web Browsers and E-mail Clients, Analyzing Webmail Services (Gmail, Yahoo Mail, Outlook.com), Investigating Social Media Activities, E-mail Spoofing and Phishing Investigations, Recovering Deleted E-mails and Web Browser History, Reporting and Presentation of Web Browser and E-mail Forensic Findings.

**Unit IX: E-mail Forensics 6 Lecture Hours**

E-mail Forensics, Introduction to E-mail Forensics, E-mail Headers and Metadata Analysis, E-mail Content and Attachments Analysis, E-mail Client Artifacts, E-mail Tracking and Tracing, E-mail Spoofing and Phishing Analysis, E-mail Authentication and Validation, E-mail Recovery and Reconstruction Techniques, Deleted E-mail Recovery, E-mail Encryption and Decryption, E-mail Timestamp Analysis, E-mail Forensic Tools and Software, E-mail Forensics in Webmail Services (Gmail, Yahoo Mail, Outlook.com), E-mail Forensics in Desktop Clients (Outlook, Thunderbird), E-mail Forensics in Mobile Devices, E-mail Forensics in Exchange Servers, Reporting and Presentation of E-mail Forensic Findings.

**Unit X: Anti-Forensics Techniques and Report Writing 6 Lecture Hours**

Anti-Forensics Techniques and Report Writing, Introduction to Anti-Forensics Techniques, Data Destruction and Deletion, Data Encryption and Steganography, Data Hiding and File Manipulation, Anti-Forensics in Network Communications, Anti-Forensics in Web Browsing and E-mails, Anti-Forensics in Mobile Devices, Anti-Forensics in Cloud Services, Anti-Forensics Countermeasures, Detecting and Overcoming Anti-Forensics Techniques, Digital Forensics Reporting, Elements of a Forensic Report, Report Structure and Formatting, Writing Clear and Concise Findings, Case Documentation and Evidence Presentation, Legal and Ethical Considerations in Report Writing, Peer Review and Quality Assurance in Report Writing.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Dr. Akashdeep Bhardwaj and Keshav Kaushik, “Practical Digital Forensics”, BPB publisher, 2023. 2. J. Bayuk, “CyberForensics: Understanding information security investigations”, Springer Science & Business Media, 2010. |
| **Reference books** | 1. E. Casey, “Handbook of digital forensics and investigation”, Academic Press, 2009. 2. E. Casey, “Digital evidence and computer crime: Forensic science, computers and the internet”, Academic Press, 2011. 3. EC-Council, “Computer forensics: Investigating network intrusions and cybercrime (CHFI)”, Cengage Learning, 2016. 4. T. J. Holt, A. M. Bossler, and K. C. Seigfried-Spellar, “Cybercrime and digital forensics: An introduction”, Routledge, 2015. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Digital Forensics Lab**

**List of Experiments**

* + - 1. Creating a Forensic Image using FTK Imager
      2. Capturing and analyzing network packets using Wireshark
      3. Data Acquisition using Autopsy Sleuthkit
      4. Forensics using Kali Linux
      5. Email Forensics
      6. Memory Forensics
      7. Malware Analysis – Dynamic
      8. Malware Analysis – Static
      9. Steganography
      10. Watermarking.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Dr. Akashdeep Bhardwaj and Keshav Kaushik, “Practical Digital Forensics”, BPB publisher, 2023. 2. J. Bayuk, “CyberForensics: Understanding information security investigations”, Springer Science & Business Media, 2010. |
| **Reference books** | 1. E. Casey, “Handbook of digital forensics and investigation”, Academic Press, 2009. 2. E. Casey, “Digital evidence and computer crime: Forensic science, computers and the internet”, Academic Press, 2011. 3. EC-Council, “Computer forensics: Investigating network intrusions and cybercrime (CHFI)”, Cengage Learning, 2016. 4. T. J. Holt, A. M. Bossler, and K. C. Seigfried-Spellar, “Cybercrime and digital forensics: An introduction”, Routledge, 2015. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSSF4023P** | **OS, Application & Cloud Security** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | **Network Security Practices -** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

The objective of this course is to provide students with a comprehensive understanding of the concepts, techniques, and best practices related to securing operating systems, applications, and cloud environments. The course aims to equip students with the knowledge and skills necessary to effectively administer and troubleshoot operating systems, manage user access and permissions, monitor system performance, and implement security configurations. Students will also learn about open-source intelligence (OSINT) and its methodologies, tools, and ethical considerations. The course further covers the fundamentals of database security, including authentication, access control, encryption, auditing, and incident response. Additionally, students will explore application security principles, secure software development practices, common vulnerabilities, and security testing techniques. The course will conclude with an examination of virtualization and cloud security, including virtual machine and hypervisor security, cloud deployment models, identity and access management, data security and privacy, incident response, and best practices in cloud security. By the end of the course, students will have the necessary knowledge and skills to effectively secure and protect operating systems, applications, and cloud environments against potential threats and vulnerabilities.

**Course Outcomes**

On completion of this course, the students will be able to

1. Understand the principles, methodologies, and techniques involved in administering and troubleshooting operating systems, including user management, file system management, process management, and system performance monitoring.
2. Develop an understanding of open-source intelligence (OSINT) methodologies, tools, and ethical considerations, enabling students to gather and analyze information from various sources for security and threat intelligence purposes.
3. Evaluate and implement database security measures, including authentication, access control, encryption, auditing, backup and recovery strategies, and vulnerability management, to protect sensitive data and ensure compliance with security standards and regulations.
4. Apply secure software development principles, including secure coding practices, threat modeling, input validation, authentication, access control, and error handling, to develop and deploy secure applications. Students will also gain proficiency in conducting security testing and code reviews to identify and remediate vulnerabilities.
5. Understand the fundamentals of virtualization and cloud computing, including virtual machine and hypervisor security, cloud deployment models, identity and access management, data security and privacy, and incident response, to effectively address security challenges and risks in virtualized and cloud environments.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 2 |
| **CO 2** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |  |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO4** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 |
| **CO5** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 |
| **Average** | **-** | **-** | **-** | 0.4 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 0.8 | 1.2 | 1 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Administration & Troubleshooting of OS 9 Lecture Hours**

Administration & Troubleshooting of OS, User Management and Access Control, File System Management, Process Management, System Performance Monitoring, System Logs and Event Viewer, Software Installation and Updates, Patch Management, Disk Management and Storage, Network Configuration and Troubleshooting, Device Drivers and Hardware Management, Backup and Recovery Strategies, OS Security Configuration, Troubleshooting Common OS Issues, System Maintenance and Optimization.

**Unit II: Open-Source Intelligence 9 Lecture Hours**

Open-Source Intelligence, Introduction to Open Source Intelligence (OSINT), OSINT Methodology and Frameworks, Information Sources and Tools, Search Engines and Advanced Search Techniques, Social Media Intelligence (SOCMINT), Online Forums and Communities, Public Records and Government Sources, Web Scraping and Data Mining, Geolocation and Mapping Tools, Metadata Analysis, Image and Video Analysis, Dark Web Intelligence, OSINT for Threat Intelligence, Ethical and Legal Considerations in OSINT.

**Unit III: Database Security 9 Lecture Hours**

Database Security, Introduction to Database Security, Database Security Threats and Risks, Database Authentication and Authorization, Access Control and Privilege Management, Data Encryption and Masking, Secure Database Design and Configuration, Database Activity Monitoring and Auditing, Database Backup and Recovery, Database Patching and Vulnerability Management, Database Security Best Practices, Database Security Standards and Regulations, Database Security Testing and Assessment, Database Security Controls and Tools, Database Security Incident Response.

**Unit IV: Application Security 9 Lecture Hours**

Application Security, Introduction to Application Security, Secure Software Development Lifecycle (SDLC), Threat Modeling and Risk Assessment, Secure Coding Practices, Input Validation and Data Sanitization, Authentication and Session Management, Access Control and Authorization, Cryptography and Key Management, Error Handling and Logging, Secure Configuration Management, Secure File and Resource Handling, Database Security and SQL Injection, Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF), Security Testing and Code Review, Web Application Firewalls (WAF), Secure Development Frameworks and Libraries, Mobile Application Security, API Security, Secure Coding Standards and Best Practices, Application Security Tools and Technologies, Application Security Incident Response.

**Unit V: Virtualization & Cloud Security 9 Lecture Hours**

Virtualization & Cloud Security, Introduction to Virtualization Technologies, Virtualization Deployment Models, Virtual Machine Security, Hypervisor Security, Virtual Network Security, Virtualization Management and Monitoring, Cloud Computing Fundamentals, Cloud Service Models (IaaS, PaaS, SaaS), Cloud Deployment Models (Public, Private, Hybrid), Cloud Security Challenges and Risks, Identity and Access Management in the Cloud, Data Security and Privacy in the Cloud, Cloud Compliance and Legal Considerations, Cloud Incident Response and Forensics, Cloud Security Best Practices.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Gerardus Blokdyk, “Mobile Operating System Security A Complete Guide”, The Art of Service - Mobile Operating System Security Publishing, 2020. |
| **Reference books** | 1. 1. Chris Dotson, “Practical Cloud Security: A Guide for Secure Design and Deployment”, O′Reilly, 2019. 2. 2. Bryan Sullivan, “Web Application Security, A Beginner's Guide”, Osborne/McGraw-Hill, 2012. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**OS, Application & Cloud Security Lab**

**List of Experiments**

1. Hardening an Operating System
2. Application Security Assessment
3. Secure Coding Practices
4. Vulnerability Scanning and Patch Management
5. Web Application Firewall (WAF) Implementation
6. Cloud Security Configuration
7. Container Security
8. Identity and Access Management (IAM)
9. Secure Authentication and Authorization Mechanisms
10. Secure Deployment and Configuration of Cloud Services

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

**6. Big Data Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSBD2011P** | **Big Data Overview and Ingestion** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Database Management Systems - CSEG2072** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the characteristics and challenges of Big Data.

2. To get familiarized with fundamentals of Big Data Ecosystem and Hadoop Architecture

3. Enable students to acquire knowledge of Data Ingestion and different Data Ingestion architectures.

4. Learn practical knowledge of applying skills and tools to ingest different types of data.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1**: Understand the concept of Data and Big Data.

**CO2**: Explain various components of Big Data Ecosystem.

**CO3**: Exploring the real-world use cases and applications of Big Data.

**CO4**: Understanding of management principles and techniques, of Big Data Ingestion

**CO5**: Infer various data ingestion architectures and apply data ingestion concepts using Sqoop & Kafka.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | 1 |
| **CO 2** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 2 |
| **CO 3** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 2 |
| **CO4** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 2 |
| **Average** | 2 | 1.4 | 1.2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.4 | 1.2 | 2 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Big Data 10 Lecture Hours**

Data Growth explosion, Categories of Data, Different Data Storage mechanisms, Introduction to a flat file, tabular and relational databases, Introduction to NoSQL data stores, Characteristics of Big data, (Volume, Velocity, Variety, Value, Veracity), Information mining and benefits of big data; Risks of Big Data, Structure of Big Data;

**Unit II: Real World Applications and Use cases**       **8 Lecture Hours**

Need for and Importance of Big Data Analytics, The Need for Standards; Case studies from various industries (e.g., finance, healthcare, e-commerce), Emerging trends and future directions in big data.

**Unit III: Big Data Ecosystem 12 Lecture Hours**

Introduction to Apache Hadoop Ecosystem and its components. Hadoop File System (HDFS); Scalable Storage, Data locality, Resilience and fault tolerance. HDFS Administration; Setup of a Hadoop Cluster; Data loading into Hadoop, Map / Reduce concepts; Managing Job Execution using Yarn; Data analysis and Visualization.

**Unit IV: Data Ingestion in Big data 10 Lecture Hours**

Different sources of Big data, Introduction to Data ingestion, Features and challenges in Big Data Ingestion, ETL and ELT, Data ingestion pipelines for Big Data, Ingesting batch vs streaming data.

**Unit V: Batch Ingestion in Big data               10 Lecture Hours**

Techniques for ingesting batch data from various sources (e.g., files, databases), Sqoop Architecture, Sqoop Import and Export, Sqoop Job, Incremental Import and File Handling.

**Unit VI: Real-Time Ingestion in Big Data 10 Lecture Hours**

Introduction to message queuing systems, Architecture of message queuing systems, Broker vs. Broker less, Distributed broker, Exchanges and Exchange Types, Distributed publish-subscribe messaging system, Kafka - Broker, Producer and consumer

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Big Data Overview – Xebia Course Material 2. Big Data Ingestion - Xebia Course Material |
| **Reference books** | 1. Sqoop: O'Reilly Media 2. The Big Data Ingestion: O'Reilly Media |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Big Data Overview and Ingestion Lab**

**List of Experiments**

**Experiment 1: Installation of the Cloudera QuickStart virtual machine**

1. Install Cloudera Hadoop Framework on the VM
2. Check the Hadoop tools present and running using HUE

**Experiment 2: Understand Basic Linux and HDFS commands**

1. Demonstrate various Linux commands
2. Demonstrate Basic HDFS commands
3. Write commands for data transfer to/from HDFS
4. Visualising the HDFS file system using HUE

**Experiment 3: Importing data using Sqoop**

1. Login to MySQL
2. Create a database **upes\_db**
3. Create table **students** in the **upes\_db** database

|  |  |  |  |
| --- | --- | --- | --- |
| **Rollno** | **Name** | **Marks** | **Subject** |
| 1 | Amit | 90 | Physics |
| 2 | Sumit | 85 | Math |
| 3 | Ram | 88 | Computer |
| 4 | Sandeep | 88 | Physics |
| 5 | John | 86 | Math |
| 6 | Tom | 85 | Computer |
| 7 | Mani | 89 | Physics |
| 8 | Akshay | 88 | Math |
| 9 | Arun | 91 | Computer |

1. Using Sqoop transfer all the data of **students** table into HDFS.
2. Using Sqoop transfer all the data of **students** table into HDFS to a specific folder **students\_1**.
3. Using Sqoop transfer all the data of **students** table who are studying **Physics** subject into HDFS to a specific folder **students\_2.**
4. Using Sqoop transfer all the data of **students** table who are studying **Computer** subject into HDFS to a specific folder **students\_pass,** without specifying the password on the command line**.**
5. Import **students** table from MySQL as a text file to the destination folder **students\_text.** Fields should be terminated by a tab character ("\t") character and lines should be terminated by newline character ("\n").

**Experiment 4: Compression and Data Format data using Sqoop**

1. Import **students** table from MySQL into HDFS to the destination folder **students\_avro**. The file should be stored as Avro file.
2. Import **students** table from MySQL into HDFS to the destination folder **students\_parquet**. The file should be stored as parquet file.
3. Import **students** table from MySQL into HDFS to the destination folder **students\_sequence**. The file should be stored as a sequence file.
4. Import **students** table from MySQL into HDFS to the destination folder **students\_compress**. Decrease the size occupied by the generated file using **GZip** codec.
5. Import **students** table from MySQL into HDFS to the destination folder **students\_mappers**. Create a number of mapper 2 and use **snappy** compression.
6. Create a new table **library** in the **upes\_db** database and import all tables from MySQL database **upes\_db** into HDFS as Avro data files use compression

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Author** |
| 1 | Java Programming | James Gosling |
| 2 | DBMS | Navathe |
| 3 | C Language | Yashwant Kanetkar |
| 4 | Big Data | Tom |
| 5 | Statistics | Atul |
| 6 | Networking | William |

**Experiment 5:** Incremental import using Sqoop

1. In Students table insert more data as below

|  |  |  |  |
| --- | --- | --- | --- |
| **Rollno** | **Name** | **Marks** | **Subject** |
| 10 | Akshat | 88 | Physics |
| 11 | Akshay | 85 | Math |
| 12 | Sunil | 87 | Computer |
| 13 | Mandeep | 84 | Physics |

1. Using Sqoop transfer only new rows of **students** table into HDFS to a specific folder **students\_increment** and verify the content at the target directory in HDFS.
2. Remove primary key from **Library** table in the upes\_db database using (alter, drop, modify) SQL commands.
3. Using Sqoop transfer all the data of the **Library** table into HDFS to a specific folder **library\_noPk** without using primary key and verify the content at the target directory in HDFS.
4. Using Sqoop do the following. Read the entire steps before you create a Sqoop job.

* create a Sqoop job Import Students table as the text file to directory **Students\_job**.
* Import all the new inserted three records to **Students** table from MySQL.
* Run the Sqoop job so that only newly added records can be pulled from MySQL.
* Validate to make sure that no duplicate records in **HDFS**

**Experiment 6:** Free form query, export and Sqoop merger

1. Create a MySQL table named **students\_1** and load data from **/home/cloudera/students\_1**

Validate to make sure the records have been added to the database.

1. Using Sqoop, import **students\_replica** table from MYSQL into HDFS such that fields are separated by a '|' and lines are separated by '\n'. Null values are represented as -1 for numbers and "NOT-AVAILABLE" for strings. Only records with roll no greater than or equal to 1 and less than or equal to 88 should be imported and use 3 mappers for importing. The destination file should be stored as a text file to directory **/home/cloudera/students\_replica**
2. Using Sqoop transfer all the data of **students** table who are studying **Physics** subject into HDFS to a specific folder **students\_query,** use SQL query to import the data.
3. Using Sqoop transfer the data of **students** table into HDFS to a specific folder **students\_m1** only student roll no less than or equal to 5, use SQL query to import the data.
4. Using Sqoop transfer all the data of **students** table into HDFS to a specific folder **students\_m2** only student roll no greater than 5.
5. Using sqoop merge data available in **students\_m1** and **students\_m2** to produce a new set of files in **students\_both.**

**Experiment 7:** Lab setup and configuration for Kakfa

1. Install JDK 8 or higher (it’s a pre-requisite)
2. Download Kafka, install and configure the properties file for Kafka and Zookeeper.
3. Installing Java 8 & IntelliJ Community Edition.

**Experiment 8 and 9:** Kafka Producer and Consumer - Command Line

1. Create a topic name first\_topic with 1-replication factor and one partition.
2. Verify if the topic is created or not.
3. Describe topic first\_topic
4. Create a topic name second\_topic with 3-replication factor and one partition ( use configuration properties file).
5. Verify all the topics
6. Describe topic second\_topic to check all ISR.
7. Send message to the first\_topic.
8. In new window read the first\_topic.
9. Read the first\_topic from the beginning.
10. Modify the first\_topic change partition to 3.
11. Delete the second\_topic

**Experiment 10:** Kafka Producer and Consumer – Java application

1. Create a Producer application using Java to implement fire and forget method of sending messages in Kafka.
2. Create the consumer application to read the message from the Producer.
3. Create a Producer application using Java to implement Synchronous way of sending messages in Kafka.
4. Create the consumer application to read the message from the Producer.
5. Create a Producer application using Java to implement Asynchronous way of sending messages in Kafka.
6. Create the consumer application or through console window to read the message from the Producer.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Kathleen Ting, and Jarek Jarcec Cecho, “Apache Sqoop Cookbook”, O'Reilly Media, 2013. 2. Neha Narkhede, Gwen Shapira, and Todd Palino, “Kafka: The Definitive Guide”, O'Reilly Media, 2017. |
| **Reference books** | 1. Sqoop: O'Reilly Media 2. The Big Data Ingestion: O'Reilly Media |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSBD3020P** | **Big Data Storage and Analysis** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Big Data Overview and Ingestion - CSBD2011P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Enable students to acquire the knowledge of NoSQL and differentiate with RDBMS.

2. Know the internal architecture of a distributed database file system.

3. To provide students with the concept of Big Data analysis and different frameworks.

4. Teach students in applying skills and tools to manage and analyze Big data.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Illustrate the file system namespace, NoSQL, and Data Lake concepts.

**CO2**. Infer Distributed file system concepts and Data Lake.

**CO3:** Define the concepts of big data analysis framework.

**CO4.** Describe data definition, description, data munging and transformation.

**CO5.** Build querying, data management, data storage using Hive and Pig.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |  | 1 | 2 |
| **CO 2** | 2 | **-** | **-** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 | 2 |
| **CO 3** | 2 | 2 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **CO4** | 2 | 2 | **-** | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **CO5** | **-** | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 |
| **Average** | 1.6 | .8 | .6 | .8 | 1.2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.2 | 1.6 | 2.4 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: The File System Namespace 14 Lecture Hours**

Traditional file systems, File system hierarchy, Limitations of traditional file systems, Introduction to NoSQL databases, Compare and contrast with RDBMS, Document-oriented, columnar, graph-based and key value pair. Scaling using sharding. Introduction to Data Lake.

**Unit II: Distributed File Systems 14 Lecture Hours**

Introduction to DFS, Client-server architecture in DFS, Cluster-based architecture in DFS, Google File System; HDFS - Block size, daemons and roles, Replication, robustness and fault tolerance, Data Disk Failure, Heartbeats and Re-Replication, High availability, Checkpointing; HBase.

**Unit III: Introduction to Big Data Analysis 12 Lecture Hours**

Requirement of Big Data analysis, Exploring Big data, mining big data, Challenges in analyzing big data. Scripting like frameworks, SQL like frameworks, Programming for Big Data analysis.

**Unit IV: Data Definition, Description, and Transformation 10 Lecture Hours**

Data type conversions, Tables, Partitions, Buckets, Views, Performing data munging, exploratory data analysis, Data transformation.

**Unit V: Common Data Analysis Frameworks   10 Lecture Hours**

Build querying, data management, data storage using Hive, Impala, and Pig

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Big Data Storage– Xebia Course Material 2. Big Data Analysis- Xebia Course Material |
| **Reference books** | 1. Hadoop: The Definitive Guide, 4th Edition - O'Reilly Media 2. MongoDB: The Definitive Guide, 2nd Edition - O'Reilly Media 3. Integrated Analytics: Courtney Webster 4. Programming Hive - O'Reilly Media 5. Programming Pig, 2nd Edition - O'Reilly Media |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Big Data Storage and Analysis Lab**

**List of Experiments**

**Experiment 1 and 2: Working with NoSQL Document DB – MONGODB**

1. To understand and perform the commands on MongoDB.

**Experiment 3: Working with HDFS**

1. Hadoop Installation and Configuration on Cloud Era
2. To understand the basic commands of Hadoop.
3. To understand the advance Hadoop commands.

**Experiment 4: HDFS Administration and Visualisation**

1. To study the HDFS Administration commands and their uses.
2. To visualise Hadoop cluser and the HDFS using HUE interface.

**Experiment 5: HBase Data Import**

To understand how HBase data import is done.

**Experiment 6: Explore HBase Shell**

1. To study and understand how to access data in HBase Shell.
2. Understanding how multiple operations on data-tables can be performed to get efficient data storage and flexible interaction to the clients.
3. Create a table, add rows, add column families, replace a row, drop table
4. retrieve a row and a cell, retrieve many rows from a given column family
5. create a new version of a value in a cell

**Experiment 7: HIVE Data Import**

To understand how HIVE data import is done.

**Experiment 8: Explore HIVE Shell – internal table**

1. create an internal relational table
2. find a location of internal relational table in HDFS
3. insert a row into an internal relational table
4. load data into an internal relational table
5. Describe and drop table

**Experiment 9: Explore HIVE Shell – external table**

1. create an external relational table
2. insert a row into an external relational table
3. load data into an external relational table
4. Describe and drop table

**Experiment 10: HIVE query language**

1. Use HQL to query the relational database
2. create and process a self-contained HQL script

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Xebia Materials 2. Tom White, "Hadoop: The Definitive Guide", 4th Edition, O'Reilly Media, 2015. |
| **Reference books** | 1. Lars George, "HBase: The Definitive Guide", O'Reilly Media, 2011. 2. Edward Capriolo, Dean Wampler, and Jason Rutherglen, "Programming Hive", O'Reilly Media, 2012. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSBD3019P** | **Big Data Processing – Disk based and In Memory** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Big Data Storage and Analysis - CSBD3020P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand Hadoop MapReduce framework and the working of MapReduce on data stored in HDFS.

2. Learning YARN concepts in MapReduce.

3. To understand In-Memory computation: Spark framework.

4. To learn the concept of RDD- Resilient Distributed Dataset, data frames, data sets, transformations, and actions.

5. To understand the working of Spark in Python and Scala.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Elaborate distributed processing framework.

**CO2.** Infer Map-Reduce architecture and build basic MapReduce programs.

**CO3**. Combine resource management framework like YARN with MapReduce.

**CO4**. Discuss Spark architecture and explain the use of core Spark APIs.

**CO5**. Analyze typical use cases of Spark.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | 2 | **-** | **-** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 3 |
| **CO4** | **-** | **-** | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 3 |
| **CO5** | 2 | **-** | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 3 |
| **Average** | 1.2 | **-** | .2 | .6 | 1.4 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | .4 | 1 | 2.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Performing distributed processing 10 Lecture Hours**

Scaling processing, Distributed processing architectures, Batch processing

**Unit II: Implementing MapReduce**      **10 Lecture Hours**

Map Reduce for distributed processing, Data locality principle, Mapper and Reducer.

**Unit III: Implementing YARN**  **10 Lecture Hours**

Introduction to YARN, Resource Manager, Node Manager, Application Master.

**Unit IV: Compare and contrast Disk Based and 10 Lecture Hours**

**In-Memory architecture**

Performance of Disk based distributed computation, Limitations of Disk-based architectures, Introduction to in-memory distributed computation, comparison of Disk-based and In-Memory processing.

**Unit V: Overview of current In-Memory technologies 10 Lecture Hours**

In-Memory architecture. Example: Spark, Driver and Worker nodes, Spark Context and Executors

**Unit VI: Concept of RDD and Working with Spark 10 Lecture Hours**

Introduction to RDD, Actions and transformations, RDD lineage, PySpark, Building with Spark API using Python, Building with Spark API using Scala .

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Big Data Storage– Xebia Course Material 2. In-Memory Computing - Xebia Course Material |
| **Reference books** | 1. Tom White, "Hadoop: The Definitive Guide", 4th Edition, O'Reilly Media, 2015. 2. Bill Chambers, and Matei Zaharia, "Spark: The Definitive Guide", O'Reilly, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Big Data Processing**  **Lab**

**List of Experiments**

**Experiment 1**

1. Write down the steps required to install and configure the MapReduce on Cloudera quickstart virtual machine.
2. Write a MapReduce job to count the numbers of words in a given file.

**Experiment 2**

1. Exploring YARN and HUE
2. Write a MapReduce job that reads any text input and computes the average length of all words that start with each character

**Experiment 3**

1. Add a Combiner in WordCount program to reduce the amount of intermediate data sent from the Mapper to the Reduce
2. Write an application that counts the number of times words appear next to each other.

**Experiment 4**

Create an application that will process a web server’s access log to count the number of times gifs, jpegs, and other resources have been retrieved. The job will report three figures: the number of gif requests, number of jpeg requests, and the number of other requests

**Experiment 5**

1. Write a MapReduce job that reads any text input and computes the average length of all words that start with each character.
2. Write a MapReduce job with multiple Reducers, and create a Partitioner to determine which Reducer each piece of Mapper output is sent to.

**Experiment 6**

1. To install spark and understand spark daemons.
2. Work with Spark Repl and Pyspark : word count problem

**Experiment 7**

1. To build page rank algorithm using spark (Python)
2. To build page rank algorithm using spark (Python)

**Experiment 8. Working with Scala Programming Language**

To create a project in Scala programming language: Word count problem,

**Experiment 9**

1. To build a simple analytical dashboard using PySpark and flask: election polling problem
2. To work with Spark GraphX

**Experiment 10**

1. To build a spark application using Python and Mysql
2. To monitor and trouble shoot spark application

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. B. Chambers and M. Zaharia, Spark: The Definitive Guide, O’Reilly. ISBN:978-93-5213-706-0 2. Hien Luu, Beginning Apache Spark 2:With Resilient Distributed Datasets, Spark SQL, Structured Streaming And Spark Machine Learning Library, APress, ISBN: 9781484235799 |
| **Reference books** | 1. Sandy Ryza, Uri Laserson, Advanced Analytics with Spark: Patterns for Learning from Data at Scale 2nd Edition, Kindle Edition, O’Reilly. ISBN:978-1-491-91276-8 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSBD4012** | **Stream Processing** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Big Data Processing - Disk based and In Memory - CSBD3019P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Explain a few concepts of Spark streaming

2. Describe basic and advanced sources

3. Explain how stateful operations work

4. Explain window and join operations

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Learn Data Streaming concepts and different frameworks.

**CO2.** Design Spark Streaming applications using the Scala/Python programming language.

**CO3.** Analyze different access log data and transform streams of it.

**CO4.** Build streaming data through stateful and stateless streaming.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | 1 | **-** |  | 2 | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 1 | 3 |
| **CO 2** | 1 |  | **-** | 1 | 2 | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 3 | 3 |
| **CO 3** | **-** | 1 | **-** | 1 | 1 | **-** | **-** |  | **-** | **-** | **-** | **-** | 2 | 2 | 1 |
| **CO4** | 1 | **-** | **-** | 2 | 2 | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 1 | 2 |
| **Average** | .5 | .5 | **-** | 1 | 1.75 | **-** | **-** | .75 | **-** | **-** | **-** | **-** | 2.25 | 1.75 | 2.25 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Overview of Stream Data Processing 14 Lecture Hours**

Batch versus real-time data processing, Understanding streams, Data storage layer, Input data streams, Resilient distributed datasets and discretized streams, Fault-tolerant Stream Processing. Benefits and challenges of Stream Processing,  stream processing frameworks.

**Unit II: Discretized Stream (Dstream) in Spark 14 Lecture Hours**

Stream Processing architecture, Programming Model – Dstream, Streaming Context, Input Dstreams and Receivers, Transformations on DStreams, Output Operations on DStreams, DataFrame and SQL Operations, MLlib Operations.

**Unit III: Structure Streaming programming 14 Lecture Hours**

Stateful streaming, Stateless streaming, Basic Operations - Selection, Projection, Aggregation, Window-based Transformations, Join Operations, Streaming Deduplication, Arbitrary Stateful Computations, Streaming Queries.

**Unit IV: Streaming Architectures 10 Lecture Hours**

Components of a Data Platform, Architectural Models, The Use of a Batch-Processing Component in a Streaming Application, Referential Streaming Architectures, The Lambda Architecture, The Kappa Architecture, Streaming Versus Batch Algorithms

**Unit V: Use case for Stream Processing 8 Lecture Hours**

Processing Distributed Log Files in Real Time, Fraud Detection, Stock Market, etc.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Stream Processing– Xebia Course Material 2. Hien Luu, “Beginning Apache Spark 2: With Resilient Distributed Datasets, Spark SQL, Structured Streaming and Spark Machine Learning Library”, Apress, 2018 |
| **Reference books** | 1. Francois Garillot, and Gerard Maas, “Stream Processing with Apache Spark”, O'Reilly Media, 2019. 2. Bill Chambers, and Matei Zaharia, "Spark: The Definitive Guide", O'Reilly Media, Inc., 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Stream Processing Lab**

**List of Experiments**

**Experiment 1:** Spark Streaming with Twitter Data

**Experiment 2:** Spark Streaming with Clickstream Data

**Experiment 3:** Integrating Spark Streaming with Spark SQL

**Experiment 4:** Integrating Spark Streaming with Apache Kafka

**Experiment 5:** Running with Spark-submit

**Experiment 6:** Implementing Spark Stateless Streaming

**Experiment 7:** Implementing Spark Stateful Streaming

**Experiment 8:** Performing Window-based Transformations

**Experiment 9:** Working with Arbitrary Combinations of Batch and Streaming Computation

**Experiment 10:** Analyzing Apache Log Files with Structured Streaming

**Experiment 11:** Streaming Applications with Different Data Formats

**Experiment 12:** ETL with Streaming Applications

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Stream Processing– Xebia Course Material 2. Hien Luu, “Beginning Apache Spark 2: With Resilient Distributed Datasets, Spark SQL, Structured Streaming And Spark Machine Learning Library”, Apress, 2018. |
| **Reference books** | 1. Francois Garillot, and Gerard Maas, "Stream Processing with Apache Spark: Mastering Structured Streaming and Spark Streaming", O′Reilly, 2019. 2. Bill Chambers, and Matei Zaharia, "Spark: The Definitive Guide", O'Reilly Media, Inc., 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSBD4011P** | **Big Data Search and Security** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | **Big Data Storage and Analysis - CSBD3020P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand Big Data Search tools: Elastic-search and Lucene

2. To Identify the key components of Lucene and setup a basic search solution

3.  Assess threats to a production Hadoop cluster, plan and deploy defenses against these threats.

4. Set up authentication and authorization with Kerberos and Apache Sentry.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.**  Discuss the big data search and analyze problems related to Elastic-search.

**CO2**. Explain the core components of Lucene and its architecture.

**CO3**. Explain data protection in distributed environments.

**CO4**. Discuss the need of securing distributed systems with Kerberos and Sentry.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 3 | 3 |
| **CO 2** | **-** | **-** | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 3 | 3 |
| **CO 3** | 2 | **-** | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 3 | 3 |
| **CO4** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **Average** | .75 | .25 | **-** | .5 | 1.5 | **-** | **-** | .25 | **-** | **-** | **-** | **-** | 1.25 | 2 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Performing exploration on massive volume of data 10 Lecture Hours**

Requirement of Big Data search, Insights using Big Data search, Text search, Big Data search tools – Elastic Search and Lucene.

**Unit II: Working with Elastic Search and Lucene 10 Lecture Hours**

Building Indexes, Document, Mapping in elastic search, Data types, Nodes and shards; Architecture of Lucene, Terms: Document, Field, Term, Token, Lucene scoring, Lucene based search tools, build a sample application using Lucene.

**Unit III: Securing Distributed Systems**  **9 Lecture Hours**

Threat Categories, Threat and Risk Assessment, Vulnerabilities

**Unit IV: Using Kerberos and Sentry with distributed systems**  **8 Lecture Hours**

Kerberos Overview, Kerberos Trusts, Kerberos Workflow, MIT Kerberos, The Sentry Service, Sentry Privilege Models, Sentry Policy Administration

**Unit V: Data protection in distributed environments 8 Lecture Hours**

Integrity of Ingested Data, Data Ingest Confidentiality, Ingest Workflows, Encryption Algorithms, Encrypting Data at Rest, Encrypting Data in Transit

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Big Data Search – Xebia Course Material 2. Big Data Security - Xebia Course Material |
| **Reference books** | 1. Andrea Gazzarini, “Apache Solr Essentials”, Packt Publishing Limited, 2015. 2. Clinton Gormle, “Elasticsearch: The Definitive Guide”, O'Reilly Media, 2015. 3. Ellen Friedman, and Ted Dunning, “Sharing Big Data Safely: Managing Data Security”, O'Reilly Media, 2016. 4. Davi Ottenheimer, “The Realities of Securing Big Data”, John Wiley & Sons, 2020. 5. Ben Spivey, and Joey Echeverria, “Hadoop Security: Protecting Your Big Data Platform”, O'Reilly Media, 2015. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Big Data Search and Security Lab**

**List of Experiments**

**Experiment No: 01 Installation**

Installing lucene and build a sample application

**Experiment No: 02 Indexing**

Indexing with local pdf files using SOLR

**Experiment No: 03 Text search**

Full text search using SOLR

**Experiment No: 04 working with data structure, Functions and methods**

Working with query dsl in elasticsearch

**Experiment No: 05** Altering apache lucene scoring

**Experiment No: 06** Access Control and Authentication Testing- kerberos

**Experiment No: 07** Data Encryption and Decryption – data at rest and in transit

**Experiment No: 08** Data Auditing and Monitoring

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Big Data Search – Xebia Course Material 2. Big Data Security - Xebia Course Material |
| **Reference books** | 1. Andrea Gazzarini, “Apache Solr Essentials”, Packt Publishing Limited, 2015. 2. Clinton Gormle, “Elasticsearch: The Definitive Guide”, O'Reilly Media, 2015. 3. Ellen Friedman, and Ted Dunning, “Sharing Big Data Safely: Managing Data Security”, O'Reilly Media, 2016. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

**7. Data Science Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDS2002P** | **Fundamentals of Data Science** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Database Management Systems -CSEG2072** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the concept of data science.

2. To understand techniques and methods related to the area of data science on real world applications.

**Course Outcomes**

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

**CO1:** Understand the fundamentals of data processing.

**CO2:** Understand and apply mathematical concepts in the field of data science.

**CO3:** Employ the techniques and methods related to the area of data science in a variety of applications.

**CO4:** Apply logical thinking to understand and solve the problem in context.

**CO5:** Apply the entire concept in data analysis tools.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 |
| **CO 2** | **-** | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 |
| **CO 3** | **-** | 3 | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 |
| **CO4** | **-** | 3 | 2 | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 |
| **CO5** | **-** | 3 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 |
| **Average** | **-** | 3 | 2 | 2.4 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2.8 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Data Science 8 Lecture Hours**

Fundamentals of Data Science, Real World Applications, Data Science Challenges, Software Engineering for Data Science (DataOps, MLOps (intro)). Data science process roles, Stages in data science.

Defining Analytics, Types of data analytics (Descriptive, Diagnostic, Predictive, Prescriptive)

Data Science Process: CRISP-DM Methodology, SEMMA, BIG DATA LIFE CYCLE, SMAM.

**Unit II: Probability and statistics for Data Science 12 Lecture Hours**

Probability: Introduction, finite sample spaces, conditional probability, independence; Random variables, distribution functions, probability mass and density functions, standard univariate discrete and continuous distributions; Mathematical expectations, moments; Random vectors, joint, marginal, and conditional distributions, independence, covariance, correlation, standard multivariate distributions, functions of random vectors; central limit theorem.

Statistics: Sampling distributions of the sample mean and the sample variance for a normal population; Point and interval estimation; Sampling distributions (Chi-square, t,F,Z), Hypothesis testing; One tailed and two-tailed tests; Analysis of variance, ANOVA, One way and two way classifications

**Unit III: Data, Data Sources and Visualization 15 Lecture Hours**

Types of Data and Datasets, Data Quality, and Issues, Data Models, General Framework of Formal modeling, Association Analyses, Prediction Analyses, Data Pipelines and patterns, Data from files & working with relational databases, Diverse data sources, data warehouses, data mining, cloud, and Data lake: Characteristics, components, Data Streaming Ingestion, Batch Data Ingestion, Data Cataloging, Data Pipeline Stages (extraction, ingestion, cleaning, exploration, wrangling, versioning, Data transformation, Feature management). Data Visualization: Overview of visualization techniques for Data Exploratory analysis

**Unit IV: Feature Engineering and Optimization 10 Lecture Hours**

Feature Extraction, Feature Construction, Feature Subset selection, Feature Learning, Feature Reduction (Dimensionality Reduction) Case Study involving FE tasks, and Feature Engineering techniques for text, images, audio, and video.Necessary and sufficiency conditions for optima; Gradient descent methods; Constrained optimization; Introduction to non-gradient techniques; Introduction to least squares optimization; Optimization view of machine learning.

**Unit V: Supervised and unsupervised learning 10 Lecture Hours**

**Introduction to Machine Learning, types, Supervised Learning: Overview, workflow, data processing, Linear Regression, Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN).**

**Unsupervised Learning: Overview, clustering algorithms: K-Means Clustering, Hierarchical Clustering, DBSCAN, Gaussian Mixture Models (GMM),**

**Dimensionality Reduction: Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE)**

**Association Rule Mining: Apriori Algorithm, FP-Growth Algorithm, Anomaly Detection, Model Evaluation (Silhouette Score, Inertia, etc.)**

**Use Cases and Practical Applications**

**Unit VI: Data Analysis Tool 5 Lecture Hours**

Reading and getting data into R, ordered and unordered factors i.e arrays and matrices – lists and data frames, reading data from files, probability distributions statistical models in R - manipulating objects – data distribution.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| Textbooks | 1. G. Strang, “Introduction to Linear Algebra”, 5th Edition, Wellesley-Cambridge Press, USA, 2016. 2. D. C. Montgomery, and G. C. Runger, “Applied Statistics and Probability for Engineers”, 5th Edition, John Wiley & Sons, Inc., NY, USA, 2011. 3. Nina Zumel, and John Mount, “Practical Data Science with R”, Manning Publications, 2014. 4. Avrim Blum, John Hopcroft, and Ravindran Kannan, “Foundations of Data Science”, 2018. Available online at: . |
| Reference books | 1. Mark Gardener, “Beginning R - The Statistical Programming Language”, John Wiley & Sons, Inc., 2012. 2. W. N. Venables, D. M. Smith and the R Core Team, “An Introduction to R”, 2013. Available online at: . 3. S. Abiteboul, R. Hull, V. Vianu, “Foundations of Databases”, Addison Wesley, 1995. 4. J. S. Bendat, and A. G. Piersol, “Random Data: Analysis and Measurement Procedures”, 4th Edition, John Wiley & Sons, Inc., NY, USA, 2010. 5. D. C. Montgomery, and G. C. Runger, “Applied Statistics and Probability for Engineers”, 5th Edition, John Wiley & Sons, Inc., NY, USA, 2011. 6. Cathy O’Neil, and Rachel Schutt, “Doing Data Science”, O’Reilly Media, 2013. |
| Web Resources |  |
| Journals |  |
| MOOCs, online courses |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Fundamentals of Data Science Lab**

**List of Experiments**

|  |  |
| --- | --- |
| **Experiment no 1** | Conduct basic data exploration by calculating summary statistics, creating histograms, and generating scatterplots. |
| **Experiment no 2** | Learn data cleaning techniques, including handling missing data, outliers, and data imputation. |
| **Experiment no 3** | Perform hypothesis tests, such as t-tests or chi-squared tests, to make inferences about data. |
| **Experiment no 4** | Implement simple linear regression to analyze relationships between variables and make predictions. |
| **Experiment no 5** | Create a variety of visualizations, including bar charts, line graphs, heatmaps, and box plots. |
| **Experiment no 6** | Use clustering algorithms to group similar data points together. |
| **Experiment no 7** | Build a random forest model for more advanced classification and regression tasks. |
| **Experiment no 8** | Discover frequent item sets and association rules in transactional data. |
| **Experiment no 9** | Project 1 (Sentiment analysis) |
| **Experiment no 10** | Project 2 (Recommendation systems) |

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. G. Strang, “Introduction to Linear Algebra”, 5th Edition, Wellesley-Cambridge Press, USA, 2016. 2. D. C. Montgomery, and G. C. Runger, “Applied Statistics and Probability for Engineers”, 5th Edition, John Wiley & Sons, Inc., NY, USA, 2011. |
| **Reference books** | 1. Mark Gardener, “Beginning R - The Statistical Programming Language”, John Wiley & Sons, Inc., 2013. 2. W. N. Venables, D. M. Smith, and the R Core Team, “An Introduction to R”, 2013. Available online at: . |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDS3005P** | **Data Visualization and Interpretation** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Python programming -CSEG1021** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Develop skills to both design and critique visualizations
2. Understand why visualization is an important part of data analysis
3. Understand the various analytical testing techniques
4. Understand the type of data impacts the type of visualization
5. Exploration and exploitation of different visualization tools.

**Course Outcomes**

Know the importance of data analytics in relation to various statistical measures.

1. Employ descriptive analytic techniques for data modelling.
2. Apply the concept of predictive analytics to various applications.
3. Design interactive dashboard for data visualization.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | **-** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 2** | 1 | **-** | **-** | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 3** | 1 | 2 | **-** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | 1 | **-** | **-** | 3 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **Average** | 1 | .5 | **-** | 1.75 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Exploratory data analysis and visualization 10 Lecture Hours**

Importance of data analysis, Descriptive vs predictive vs prescriptive, Overview of commonly used techniques, Univariate analysis, Distributions, Outlier detection, preprocessing of data, Histograms, Kernel density estimate plots, Box and violin plots, Regression plots, Bar charts, Classification, Association and Segmentation models using Python/R- programming.

**Unit II: Exploratory Data Analytics and Charts  15 Lecture Hours**

T-Test, Bi-variate Analysis, Correlations, Crosstabs, Heatmaps, Clustered Matrices Stacked bars, Line charts, Multi-variate Analysis, Trending analysis, Geographical analysis, Maps, Scatter plot, Multi line charts, Area graph, Labs using R/Python.

**Unit III: Hypothesis Testing  10 Lecture Hours**

Hypothesis Testing, T-test, Chi-square test, ANNOVA, Clustering Analysis, Cognitive Analytics, create reports based on relationships, Labs using R/Python.

**Unit IV: Visualizing Business Intelligence  10 Lecture Hours**

Representing data using graphs and charts- Line chart, Bar, Box plot, Scatter chart, building visualization reports using tableau

**Unit V: Dashboard and Scorecards  10 Lecture Hours**

Scorecards and Dashboard creation for data analytics, Dynamic reporting, enhance user interaction using Cognos Analytics

**Unit VI: Case Studies 5 Lecture Hours**

Use case studies on real data: Stock prices prediction, Stock market analysis, Election poll analysis.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. Cole Nussbaumer Knaflic, “Storytelling with Data: A Data Visualization Guide for Business Professionals”, Wiley, 2015. 2. 2. , “Data Visualization: A Practical Introduction”, , 2018. |
| **Reference books** | 1. Steve Wexler, Jeffrey Shaffer, and Andy Cotgreave, “The Big Book of Dashboards”, Wiley, 2017. 2. Stephen Few, “Information Dashboard Design”, Analytics Press, 2013. 3. Colin Ware, “Information Visualization: Perception for Design”, 4th Edition, Morgan Kaufmann Publishers In, 2020. 4. Kieran Healy, “Data Visualization: A Practical Introduction”, Princeton University Press, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Data Visualization and Interpretation Lab**

**List of Experiments**

|  |  |
| --- | --- |
| **Experiment no 1** | Introduction to Cognos Insight |
| **Experiment no 2** | Personal Analysis |
| **Experiment no 3** | What is analysis |
| **Experiment no 4** | Data Cube and dimension creation |
| **Experiment no 5** | Pivot table creation |
| **Experiment no 6** | Introduction to Plotly to visualize data |
| **Experiment no 7** | Identification of various types of data in real life |
| **Experiment no 8** | Analysing Panel data |
| **Experiment no 9** | Project 1 (Time series analysis) |
| **Experiment no 10** | Project 2 (Cross sectional data analysis) |

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. Cole Nussbaumer Knaflic, “Storytelling with Data: A Data Visualization Guide for Business Professionals”, Wiley, 2015. 2. 2. , “Data Visualization: A Practical Introduction”, , 2018. |
| **Reference books** | 1. Steve Wexler, Jeffrey Shaffer, and Andy Cotgreave, “The Big Book of Dashboards”, Wiley, 2017. 2. Stephen Few, “Information Dashboard Design”, Analytics Press, 2013. 3. Colin Ware, “Information Visualization: Perception for Design”, 4th Edition, Morgan Kaufmann Publishers In, 2020. 4. Kieran Healy, “Data Visualization: A Practical Introduction”, Princeton University Press, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDS3006P** | **Machine Learning and Deep Learning** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Elements of AIML - CSAI2018** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the fundamental concepts and usage of machine learning.
2. To learn ML algorithms to solve various tasks in real life.
3. To understand the need and use of Deep Learning.

**Course Outcomes**

On completion of this course, the students will be able to

**CO 1.** Understand types of Machine Learning algorithms, their usage and evaluation.

**CO 2.** Use Regression for univariate and multivariate analysis.

**CO 3**. Apply classification and clustering techniques.

**CO 4.** Understand Deep Learning Techniques and their applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 2** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO4** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 2 |
| **Average** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 2 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction 10 Lecture Hours**

The Origins of Machine Learning, Uses and Abuses of Machine Learning, Generalization, Bias and variance, Overfitting vs. Underfitting, Assessing the Success of Learning, Accuracy, Precision, Recall, F-1 Score, Machine Learning Pipeline, Features- selection, reduction, enhancement. Types of Machine Learning Algorithms- Supervised, Unsupervised, Semi-supervised, Active, Incremental, Multi-modal, and Transfer learning. Matching Data to an Appropriate Algorithm.

**Unit II: Regression 10 Lecture Hours**

Regression basics: Relationship between attributes using Covariance and Correlation Relationship between multiple variables, Linear Regression, Multiple Linear Regression, Polynomial Regression. Regularization methods- Lasso, Ridge, and Elastic nets, of error measures (ROCR).

**Unit III: Classification 15 Lecture Hours**

Probabilistic classifiers and Deterministic Classifiers, Binary and Multi-class Classification, Multi-label Classification, Logistic Regression, Decision Trees, Naive Bayesian Classifier, k-Nearest Neighbor Algorithm, Support Vector Machines, Linear Discriminant Analysis, Ensemble Methods: Bagging, Boosting, Random Forests, Advanced Classification Methods.

**Unit IV: Clustering 10 Lecture Hours**

Unsupervised Learning, Cluster Analysis, Partitioning Methods- K-Means, K-Medoids, Spectral Clustering. Hierarchical Methods- Agglomerative, Divisive, BIRCH. Density-Based Methods- DBSCAN, OPTICS.

**Unit V: Deep Learning 15 Lecture Hours**

Neural Networks- Perceptron, Back Propagation. Deep Networks- Definition, Motivation, Applications, Restricted Boltzmann Machine, Sparse Auto-encoder, Deep Belief Net, Hidden Markov Model. Convolution Neural Network (CNN)- Basic architecture, Activation functions, Pooling, Handling vanishing gradient problem, Dropout, Greedy Layer-wise Pre-training, Weight initialization methods, Batch Normalization. Different CNN Models- Alex Net, VGG Net, Google Net, Res Net, Dense Net. Graphical Model- Bayes Net, Variational Auto-encoders. Recurrent Neural Network (RNN), Gated RNN, Long short-term memory (LSTM).

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Tom M. Mitchell, “Machine Learning”, McGraw Hill Education, 2017. 2. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2016. |
| **Reference books** | 1. I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning”, MIT Press, 2016. 2. , “Artificial Intelligence, Machine Learning, and Deep Learning”, Mercury Learning & Information, 2020. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Machine Learning and Deep Learning Lab**

**List of Experiments**

|  |  |
| --- | --- |
| **Experiment no 1** | Write a program to find Precision, Accuracy, Recall & F1 Score. |
| **Experiment no 2** | Write a program to perform Linear Regression. |
| **Experiment no 3** | Write a program to perform Multiple Linear Regression. |
| **Experiment no 4** | Write a program to showcase Lasso, Ridge and Elastic Net Regularization. |
| **Experiment no 5** | Write a program to perform Logistic Regression. |
| **Experiment no 6** | Write a program to perform classification using k-NN. |
| **Experiment no 7** | Write a program to perform classification using Support Vector Machine. |
| **Experiment no 8** | Write a program to perform classification using Linear Discriminant Analysis. |
| **Experiment no 9** | Write a program to perform k-Means Clustering. |
| **Experiment no 10** | Write a program to build a simple neural network. |
| **Experiment no 11** | Write a program to build a Convolutional Neural Network. |
| **Experiment no 12** | Write a program to build an LSTM model. |

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Aurélien Géron, “Hands-on ML with Scikit-Learn, Keras & TensorFlow”, 3rd Edition, O’Reilly Media, 2022. 2. , , “Machine Learning and Deep Learning Using Python and TensorFlow”, McGraw Hill, 2021. |
| **Reference books** | 1. Seth Weidman, “Deep Learning from Scratch: Building with Python from First Principles”, Shroff/O'Reilly, 2019. 2. , , , and , “Machine Learning with PyTorch and Scikit-Learn”, , 2022. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSEG4040P** | **Computational Linguistic and Natural Language Processing** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Applied Machine Learning -** **CSAI2017P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand fundamentals of linguistics and language processing.
2. To study various approaches to construct a natural language processing system like information retrieval system can be developed

**Course Outcomes**

On completion of this course, the students will be able to

**CO1.** Explain basic concepts of natural language processing principles and approaches.

**CO2**. Demonstrate steps involved in development of information retrieval system with advancements.

**CO3**. Discuss the emerging applications of NLP.

**CO4.** Discuss the empirical, statistical, and classical methods in NLP and its usage.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 2** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 3** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **Average** | 1 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Classical Approaches of NLP 10 Lecture Hours**

Introduction, Classical approaches to natural language processing, Approaches to natural language processing, understanding linguistics, Level 1: Morphology, Level 2: Syntax, Level 3: Semantics, Level 4: Pragmatics, Understanding linguistics, Traditional approach, Example: Automatic summarization using NLP, Drawbacks, Text processing, What Is text processing? Text analysis vs. Text mining vs. Text analytics, Tools and methodologies: Statistical methods, Tools and methodologies: Text classification, Tools and methodologies: Text extraction, Tools and methodologies: Example, Scope of text analysis/processing, Importance of text analysis, Working principles of text analysis, Data gathering, Data preparation, Data preparation steps, Data analysis, Evaluation of text classification process, Text extraction, Analysis in test extraction, Evaluation of text extraction process, Text analysis APIs, Levels of NLP, Lexical analysis, Pre-processing activity, POS tagging, Syntactic parsing, Types of parsing, Derivation logic, Grammar, Semantic analysis, Semantic analysis elements, Representation in semantic analysis, Natural language generation, NLP vs NLG, History of NLG, Working principle of natural language generation, Limitations in natural language generation.

**Unit II: Empirical Approaches 10 Lecture Hours**

Corpus creation, Corpus linguistics, Types of corpora, Lexicographical implementations in corpora, Timeline of corpus linguistics, Usage areas of corpora, Traits of a good text corpus, Annotations in text corpus, NLP task-specific training corpora, Data sets used for natural language processing, Treebank annotation, Linguistic description layers, Areas using text annotations, Usage of annotations and corpora, Kinds of annotations, Annotation semantic labels, Annotations in machine learning, Annotation development cycle, Model creation, Create annotations, Training and testing the algorithms, Result evaluation, Revision of the model, Tree banks and its construction, Need for tree bank, Types of tree bank corpus, Phrase structured vs dependency structured tree bank, Fundamental statistical techniques, Problems of the traditional approach, How statistics helps, Problems of the traditional approach and how statistics helps, Hidden Markov model, Maximum entropy Markov model, Conditional random field model, Support vector machine, N-GRAM, Genetic algorithm, POS Tagging, Word sense disambiguation, POS tag and Tagsets, Types of POS taggers, Markovian model, Hidden Markov model, POS tagging using HMM.

**Unit III: Statistical Approaches 15 Lecture Hours**

Parsing, Statistical parsing, Approaches to parsing, Statistical approach, Lexicalized statistical parsing, Top-down parsing, Bottom-up parsing, Left corner parsing method, Statistical parsing: Probabilistic parser, Multiword expressions, Features of MWE, Types of multi word expressions, Multi word verbs, Word similarity and text similarity, Text similarity methods, Jaccard similarity, K-means, Cosine similarity, Word Mover’s distance, Variational auto encoders, Pre-trained sentence encoders, Bidirectional Encoder Representations from Transformers (BERT) with cosine distance, Word sense disambiguation, Complications in WSD, Methods in WSD, Evaluation of WSD, History of speech recognition technology, Working principle in voice recognition, Major leaders in speech recognition and voice assistant, Amazon Alexa, Microsoft Cortana, Google Assistant, Machine translation, Rule-based machine translation, Statistical machine translation, Rule-based MT vs. statistical MT, Working principle of SMT, Challenges with statistical machine translation.

**Unit IV: Applications of Natural Language Processing 15 Lecture Hours**

Information retrieval, Information retrieval in NLP, IR development, Model types, Model types: Mathematical basis model, Problems with NLP in information retrieval, NLP in information retrieval, IR evaluation metrics, Information Retrieval (IR) model and types, Design features of IR systems, Design features of IR systems, Question answering systems, QA system architecture, QA system types, Text based QA systems, Factoid question answering system, Web based question answering system, Information retrieval or information extraction based QA systems, Restricted domain question answering, Rule based question answering systems, Information extraction, Working of information extraction, Information extraction applications, Chunking, Representing chunks: Tags vs trees, Report generation, Text report specifications, Features of reports, Report generation process, Usage of NLP text in report generation, Ontology construction, Ontology classifications and process, Why ontology and its advantages, Ontology components, Levels of formality, Ontology construction approaches, Ontology construction.

**Unit V: Emerging Applications of Natural Language 10 Lecture Hours**

**Generation in Information Visualization, Education, and Health Care**

Multimedia presentation generation, Focus points to add multimedia in NLG, Text generation: Meaning representation, Text generation: Document structure design, Text generation: Linguistic style control, Document layout, Layout and meaning representation, Layout style and wording representation, Image style and meaning representation, Image and wording usage, Scripted dialogue, Language interfaces for intelligent tutoring systems, CIRCSIM-Tutor, CIRCSIM-Tutor architecture, data presentation and process cycle, AUTOTUTOR, AUTOTUTOR architecture and process, ATLAS Andes, Andes system architecture and design, Pedagogical considerations in Andes, WHY2-ATLAS, Why 2 Atlas architecture and process, Argumentation for healthcare consumers, CDS architecture and processing, NLP for CDS scope, NLP models, Building blocks of NLP - CDS, Data based evidence collection: Summarization, Applications of NLP in healthcare, Sentiment analysis and subjectivity, Difficulties in sentiment analysis, Document level sentiment classification, Sentence level sentiment classification, Lexicon, Feature-based sentiment analysis, Opinion summarization.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. . Christopher D. Manning, and Hinrich Schutze, “Foundations of Natural Language Processing”, 6th Edition, The MIT Press Cambridge, Massachusetts London, England, 2003.  2. Daniel Jurafsky and James H. Martin “Speech and Language Processing”, 3rd Edition, Prentice Hall, 2009. |
| **Reference books** | Nitin Indurkhya, and Fred J. Damerau “Handbook of Natural Language Processing”, 2nd Edition, CRC Press, 2010.  2. James Allen, “Natural Language Understanding”, 8th Edition, Pearson Publication, 2012.  3. Christopher D. Manning and Hinrich Schütze, “Foundations of Statistical Natural Language Processing”, 2nd edition, MIT Press Cambridge, MA, 2003.  4. Hobson Lane, Cole Howard, and Hannes Hapke, “Natural language processing in action”, Manning Publications, 2019.  5. Alexander Clark, Chris Fox, and Shalom Lappin, “The Handbook of Computational Linguistics and Natural Language Processing”, Wiley-Blackwell, 2012. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Computational Linguistics and Natural Language Processing Lab**

**List of Experiments**

**EXPERIMENT-1:**

Title: Installing various packages required for analytics in python and write code in Python for following program

1. Write code to load CSV file containing information about employee of a company in python and draw graph showing average salary department wise.

**EXPERIMENT-2:**

Title: Text Retrieval

1. Connect to Twitter account and Extract first 100 tweets from it in a file.
2. Study and Implementation of Processing text(Word and Sentence Tokenization)

**EXPERIMENT-3:**

Title: Processing Data

1. Python code to read a text document and perform basic pre-processing techniques on the text like tokenization, stop-word-removal, lemmatization etc..
2. Study and Implementation of Morphological analysis.

**EXPERIMENT-4:**

Title: Do text mining on extract data and Accessing text corpus

1. Calculate word count of a given specific document and Show top 10 frequent words with their frequency and Create world cloud and show graphically.
2. Study and Implementation of NER (Name Entity Recognition)

**EXPERIMENT-5:**

Title: POS-Tagging and Tagging and Parsing

1. Categorizing and tagging words in Twitter Data.
2. Study and implementation of POS Tagging and Chunking in a sentence

**EXPERIMENT-6:**

Title: Language Processor

1. Implement N–Gram Language Mode and Smoothing.

**EXPERIMENT-7:**

Title: Do sentimental analysis

1. Analysis of Sentiment and Subjectivity
2. Implement sentimental analysis on IMDB Movie Reviews Dataset.

**EXPERIMENT-8:**

Title: Do Text Summarization

1. Analysing Meaning of Sentences
2. Implement Text Summarization on IMDB Movie Reviews Dataset.

**EXPERIMENT-9 & 10:**

Title: Mini-Project on NLP

Implement Mini-Project on NLP applications.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Daniel Jurafsky, and James H. Martin, “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”, 2nd Edition, Pearson, 2013. |
| **Reference books** | 1. David A. Grossman, and Ophir Frieder, “Information Retrieval: Algorithms and Heuristics”, Springer, 2004. 2. Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, “Natural Language Processing: A paninian perspective”, Prentice Hall, New Delhi, 1995. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSDS4004P** | **Generative Artificial Intelligence** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 04** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | **Machine learning and deep learning -CSDS3006P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1.       To introduce the Generative models.

2.       To enrich the knowledge in various methods of GenAI and their application in different research applications.

3.       To extend the knowledge of GenAI to solve the current problems of AI.

4.       To provide practical programming skills necessary for constructing a GenAI.

**Course Outcomes**

On completion of this course, the students will be able to

**CO 1.** Comprehend different GenAi techniques.

**CO 2.** Use concepts of GenAI to different applications and areas.

**CO 3**. Builds new GenAI method with explainability.

**CO 4.** Aware from responsibilities and social impact of GenAI.

**CO 5.** Build GenAi methodology to improve the performance of any problem area.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 2** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO 3** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |
| **Average** | **-** | 2 | 2 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: 15 Lecture Hours**

Introduction to GenAI, Encoder-decoder, Autoencoder, Variational autoencoder, Txt2Txt translation, Txt2Img translation, Img2Txt translation, Txt2Voice translation

**Unit II: 15 Lecture Hours**

Text summarization, Text translation, Question-answer bots, Generative adversarial network, Auto-regressive methods, Attention mechanism, Soft attention, Hard attention, Self-attention

**Unit III: 15 Lecture Hours**

Transformers and its variants, Large language models, GPT-3, ChatGPT, Prompt engineering, Introduction to LangChain

**Unit IV: 15 Lecture Hours**

Explainability, Responsibility, Ethical AI, DeepFake,

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning”, MIT Press, 2016. 2. , and , “GANs in Action Deep Learning with Generative Adversarial Networks”, Manning, 2019. |
| **Reference books** | 1. Sudharsan Ravichandran, “Hands on Deep learning algorithms with python”, Packt Publishing Limited, 2019. 2. , “Generative Adversarial Networks with Industrial Use Cases”, BPB Publisher, 2020. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Generative Artificial Intelligence Lab**

**List of Experiments**

|  |  |
| --- | --- |
| **Experiment no 1** | WAP to create an autoencoder for structured data. |
| **Experiment no 2** | WAP to create an autoencoder for Txt2Txt translation. |
| **Experiment no 3** | WAP to create variational autoencoder for Img2Txt translation. |
| **Experiment no 4** | Develop a GenAI methodology for Text summarization. |
| **Experiment no 5** | Develop GAN method to augment images. |
| **Experiment no 6** | Develop GAN variant method for language translation |
| **Experiment no 7** | WAP to use different attention mechanism for image recognition |
| **Experiment no 8** | Compare various attention mechanism performance in language translation |
| **Experiment no 9** | Develop a Transformer method for time series data |
| **Experiment no 10** | Develop transformer method to analyze signal data |
| **Experiment no 11** | Develop a DeepFake method for images |
| **Experiment no 12** | Develop a DeepFake method for video data. |

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning”, MIT Press, 2016. 2. , “Hands-On Generative Adversarial Networks with Keras”, , 2019. |
| **Reference books** | 1. Sudharsan Ravichandran, “Hands on Deep learning algorithms with python”, Packt Publishing Limited, 2019. 2. , “Generative Adversarial Networks Cookbook”, Limited, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

**8. IOT Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSIS2013P** | **Introduction to IoT, Sensors and Microcontrollers** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Digital Electronics - ECEG1012** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the basics of Internet of Things
2. Learn to program the micro controllers boards
3. Understand the working of sensors and programming them
4. Implement the knowledge on a real world project

**Course Outcomes**

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

**CO1.** Students will be able to understand the fundamental concepts and principles of the Internet of Things (IoT) and its applications.

**CO2.** Students will be able to demonstrate proficiency in programming microcontrollers and interfacing sensors for IoT applications.

**CO3.** Students will be able to design and develop IoT applications using appropriate hardware and software components.

**CO4.** Students will be able to apply principles of sensor instrumentation and data acquisition for IoT systems.

**CO5**. Students will be able to analyze and solve challenges in IoT deployment, security, and scalability.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** |  | 1 |
| **CO4** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | 2 | 1 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | 2 | 2 |
| **Average** | **-** | **-** | **-** | **-** | 1.2 | **-** | **-** | **-** | .8 | **-** | **-** | **-** | **-** | 1.6 | 1.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: IoT Introduction 12 Lecture Hours**

1. Concepts and Definitions of The Internet of Things (IoT).
2. History of IoT
3. IoT Market Study
4. Requirements Functionalists and structure of IoT.
5. IoT enabling technologies.
6. IoT Architecture.
7. Major component of IoT (Hardware & Software).
8. Challenges in IoT

**Unit II: IoT Programming Platforms 12 Lecture Hours**

1. Microcontrollers Basics
2. Introduction to Arduino and various variants
3. Overview of Breadboard, Jumper Wires, LEDs, LCD, Male header Pins, Potentiometers, Buttons
4. Arduino Pin Description
5. Digital vs Analog vs PWN Pins
6. Arduino Programming basics
7. Blinking LED and playing Buzzer
8. Communication with SPI and I2C
9. Displaying on LCD and OLED

**Unit III: Sensors and Actuators with Arduino 10 Lecture Hours**

1. Temperature and Humidity Sensor, Distance Ranging Sensor
2. Acceleration Sensor, Tilt Switch Module, SD Card Module, JoyStick, RF Module, BT Module
3. IR and PIR Sensor, Force Sensors, Gas Sensor, Water Flow sensor, Sound sensor module
4. Medical Sensors: ECG, Pulse and Heart Rate Sensors, Fingerprint, Touch,
5. Soil and pH sensor
6. GPS Module and GSM Module
7. Servo Motor, DC Motors programming with Arduino

**Unit IV: Sensor and Instrumentation**   **14 Lecture Hours**

1. Diode, Resistors, Capacitors, Transistor
2. Data Acquisition System
3. Sensor Classifications
4. Sensor Characteristics:- Transfer function, Calibration, Accuracy, Reliability
5. Physical Principle of Sensing: Capacitance, Magnetism, Resistance, Induction, Piezo electric effect, Hall effect, Light
6. Dynamic Models of Sensor Elements

**Unit V: Popular Microcontrollers for IoT 12 Lecture Hours**

1. ESP32 Introduction and variants
2. ESP32 Pin Desciption
3. ESP32 as AP, WebServer
4. ESP32 connection to Internet with MQTT
5. Raspberry Pi Introduction and variants
6. Arduino vs Raspberry Pi vs ESP32
7. Cloud Platforms for IoT:- ThingSpeak, AWS IoT Core, Azure IoT Hub, GCP IoT Core

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. Jermy Blum, “Exploring Arduino”, Wiley, 2013. 2. 2. Sudip Misra, Anandarup Mukherjee, and Arijit Roy, “Introduction to IoT”, Cambridge University Press, 2022. 3. 3. D. Patranabis, “Sensors and Transducers”, 2nd Edition, PHI Learning, 2003. |
| **Reference books** | 1. Jacob Fraden, "Handbook of Modern Sensors: Physics, Designs, and Applications”, Springer, 4th Edition, 2014. 2. Dr Kamlesh Lakhwani, Dr Hemant Kumar Gianey, Joseph Kofi Wireko, and Kamal Kant Hiran, “Interent of Things”, BPB Publisher, 2020. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Introduction to IoT, Sensors and Microcontrollers Lab**

**Course Objectives**

1. Understand the basics of Internet of Things

2. Learn to program the micro controller’s boards

3. Understand the working of sensors and programming them

4. Implement the knowledge on a real world project

**Course Outcomes**

**CO1.** Students will be able to understand the fundamental concepts and principles of the Internet of Things (IoT) and its applications.

**CO2.** Students will be able to demonstrate proficiency in programming microcontrollers and interfacing sensors for IoT applications.

**CO3**. Students will be able to design and develop IoT applications using appropriate hardware and software components.

**CO4**. Students will be able to apply principles of sensor instrumentation and data acquisition for IoT systems.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 |
| **CO4** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 |
| **Average** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | .5 | **-** | **-** | **-** | **-** | 1.5 | 1.5 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**List of Experiments**

**Experiment 1: Introduction to Arduino:**

* Arduino IDE Installation.
* Setting up an Arduino board, writing a simple program to blink an LED, and understanding the basics of the Arduino IDE.
* Blink the inbuilt IDE with frequency as per the last two digit of your rollNo.
* Write the Arduino code to read a number from the serial monitor and display the inbuilt LED equally number of times.

**Experiment 2: Sensor Interfacing:**

* Write the Arduino code to read potentiometer value from the analog pins and display it on the Serial Monitor.
* Interfacing a temperature and humidity sensor (DHT11) with the Arduino board to measure the temperature and humidity.
* Interface the ultrasonic sensor(US) with Arduino and read the distance.
* WAP in Arduino to play a buzzer if the distance read from the US sensor is less than 50cm

**Experiment 3: Display Data on LCD, OLED:**

* Using a Passive Infrared (PIR) sensor to detect motion and trigger an output, such as an LED or buzzer.
* Implementing an Ultrasonic sensor to measure distance and display the results on an LCD.
* Interface the OLED display with Arduino and display your name or rollNo.
* Interface the DHT11 sensor with Arduino and display the temperature and humidity on OLED display.

**Experiment 4: IoT Basic Communication:**

* Establishing communication between two Arduino boards using the SPI or I2C protocol.
* Data Logging with SD Card: Collect any sensor data and store it on an SD card for future analysis.

**Experiment 5: Data Processing:**

* Use the Joystick to move any alphanumeric character over the OLED display.
* Read 10 consecutive readings from the US Sensor and display basic statistics on it over the OLED display.
* Display moving average of 10 consecutive readings from the US sensor over the OLED display.

**Experiment 6: GPS Module:**

* Interface a GPS module with Arduino, read GPS data such as latitude and longitude, and display the data on an LCD screen or serial monitor.
* Utilize a GPS module and Arduino to measure the speed of a moving object or vehicle by calculating the distance traveled over time using GPS coordinates. The speed data can be displayed on an LCD or sent wirelessly for further analysis.

**Experiment 7: Bluetooth Communication:**

* Interface a Bluetooth module with the Arduino board and transfer data to a PC.
* With the help of Bluetooth module control the brightness of the Arduino inbuilt LED.

**Experiment 8: Medical Sensor:**

* Write Arduino code to read PPG data and calculate the heart beat over a period of 3 minutes.
* Write Arduino code to read ECG data and calculate the heart beat over a period of 3 minutes.

**Experiment 9: Motor Control:**

* Control the movement of a servo motor with Arduino. Rotate the servo motor left by 45 degree and right by 90 degree and repeat the pattern.
* With the help of motor driver shield, interface and control the speed of the DC motor by entering various values through serial monitor.

**Experiment 10: RFID Tag Module:**

* Read and display the unique identification (UID) of RFID tags using an Arduino and an RFID module.
* For the use case of an inventory management system using Arduino and RFID technology. Use RFID tags to label and track few items. The Arduino can read the RFID tags and update the inventory database accordingly, providing real-time information on item availability and location.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Jermy Blum, “Exploring Arduino”, Wiley, 2013. 2. Sudip Misra, Anandarup Mukherjee, and Arijit Roy, “Introduction to IoT”, Cambridge University Press, 2022. 3. D. Patranabis, “Sensors and Transducers”, 2nd Edition, PHI Learning, 2003. |
| **Reference books** | 1. Jacob Fraden, "Handbook of Modern Sensors: Physics, Designs, and Applications”, Springer, 4th Edition, 2014. 2. Dr Kamlesh Lakhwani, Dr Hemant Kumar Gianey, Joseph Kofi Wireko, and Kamal Kant Hiran, “Interent of Things”, BPB Publisher, 2020. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSIS3024P** | **IoT Network Architecture and Communication Protocols** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Introduction to IoT, Sensors and Microcontrollers - CSIS2013P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand IoT Network Architectures.
2. Explore Communication Protocols.
3. Implement Network Security Measures:
4. Ensure Interoperability and Integration:

**Course Outcomes**

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

**CO1.** Students will be able to understand the fundamental principles and concepts of IoT network architecture.

**CO2.** Students will gain knowledge of various communication protocols and wireless protocols in IoT and effectively select and utilize the appropriate protocols for different IoT applications.

**CO3.** Students will be able to demonstrate proficiency in configuring and managing IoT networks, including network monitoring, diagnostics.

**CO4.** Students will be able to evaluate and implement appropriate security measures and protocols to safeguard IoT networks from potential threats and vulnerabilities.

**CO5.** Students will be able to explore IoT integration and interoperability concepts enabling seamless data exchange and interoperability between diverse IoT devices and systems.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** |  | 1 |
| **CO4** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | 2 | 1 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | 2 | 2 |
| **Average** | **-** | **-** | **-** | **-** | 1.2 | **-** | **-** | **-** | .8 | **-** | **-** | **-** | **-** | 1.6 | 1.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to IoT Network Architecture 12 Lecture Hours**

1. Overview of Computer Network
2. OSI Model Brief
3. IoT Reference Model by IoT World Forum
4. Serial vs Parallel communication
5. SPI, I2C, USB, CAN, RS232 Interfaces
6. Concepts and fundamentals of IoT network architecture
7. IoT network topologies and architectures
8. Edge computing and fog computing in IoT
9. IoT network scalability and manageability
10. Security considerations in IoT network architecture

**Unit II: IoT Communication Protocols 12 Lecture Hours**

1. IoT Data Protocols and Formats
   1. JSON (JavaScript Object Notation) for IoT data exchange
   2. XML (eXtensible Markup Language) for IoT data representation
   3. CBOR (Concise Binary Object Representation) for efficient IoT data encoding
2. MQTT (Message Queuing Telemetry Transport) protocol
3. CoAP (Constrained Application Protocol) for IoT
4. AMQP (Advanced Message Queuing Protocol)
5. HTTP and RESTful APIs in IoT
6. WebSocket protocol for real-time communication
7. Modbus protocol for industrial IoT communication

**Unit III: Wireless Communication Protocols 12 Lecture Hours**

1. Physical Layer, Modulation and demodulation
2. Radio Frequency Spectrum
3. RF Spectrum for communication
4. Transmitting data with radio waves
5. Signal distortion and noise
6. Bluetooth and Bluetooth Low Energy (BLE) in IoT
7. Zigbee protocol for low-power wireless networks
8. LoRaWAN (Long Range Wide Area Network) for IoT
9. Medium Access Control layer
10. IEEE 802.15.4 Protocol

**Unit IV: IoT Security and Privacy Protocols 12 Lecture Hours**

1. Authentication and access control in IoT
2. Encryption and secure communication protocols
3. DTLS (Datagram Transport Layer Security) for IoT
4. IPSec (Internet Protocol Security) for secure IoT communication

**Unit V: IoT Network Management and Monitoring 12 Lecture Hours**

1. SNMP (Simple Network Management Protocol) in IoT
2. IoT device provisioning and management
3. Over-the-Air (OTA) updates for IoT devices

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. MIsha Dohler, "Internet of Things: Architectures, Protocols, and Standards". 2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, and Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, 2017. |
| **Reference books** | 1. Yan Zhang, Laurence T. Yang, and Huansheng Ning, "Wireless Communications and Networks for the Internet of Things", Auerbach Publications, 2019. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**IoT Network Architecture and Communication Protocols Lab**

**List of Experiments**

**Experiment 1: Introduction to ESP8266 family of microcontrollers:**

* ESP8266 Architecture and Pin Description
* ESP32, ESP8266 variants
* Using ESP32 scan all the Wi-Fi network available and print their Wi-Fi signal strength and other information.

**Experiment 2: Sensor Interfacing:**

* Write the ESP32 code to read potentiometer value from the analog pins and display it on the Serial Monitor.
* Interfacing a temperature and humidity sensor (DHT11) with the ESP32 to measure the temperature and humidity.
* Interface the ultrasonic sensor (US) with ESP32 and read the distance.

**Experiment 3: ESP8266 Web Server:**

* Create a webserver with ESP8266 with login functionality.
* Create ESP8266 as AP and connect 2-3 WiFi devices to it.
* WAP to display WiFi signal strength on a OLED display using ESP8266.

**Experiment 4: IoT Data Protocol:**

* Connect two ESP8266 boards and transfer sensor data from one to the other using JSON format.
* Connect two ESP8266 boards and transfer sensor data from one to the other using XML format.

**Experiment 5: Wireless control:**

* Write the Arduino code for ESP8266 to measure soil moisture and pH levels using a soil moisture sensor and a pH sensor.
* Write the Arduino code for ESP8266 to rotate the servo motor by ‘n’ degree through the web interface.

**Experiment 6: UDP Protocol:**

* Set up the ESP32 as a Wi-Fi access point and configure it to act as a UDP server. Create a UDP socket on the ESP32 and bind it to a specific port for incoming data. In the main loop, continuously listen for UDP packets from connected clients. Upon receiving a UDP packet, extract and process the data.

**Experiment 7: ZigBee Protocol:**

* Understand ZigBee Protocol stack and architecture and the boards and devices available.
* Set up a simple wireless communication system using Zigbee modules. Connect two Arduino boards with Zigbee modules and program them to establish a wireless communication link. Send and receive data between the two modules to demonstrate the basic functionality of the Zigbee protocol.

**Experiment 8: ZigBee Networking:**

* Create a sensor network using Zigbee modules and Arduino boards. Connect multiple sensor nodes to an Arduino board, each equipped with a Zigbee module. Program the nodes to collect data from sensors and transmit it to a central coordinator node. Receive and process the sensor data at the coordinator node to showcase the ability of Zigbee to form a robust and scalable network.
* Build a Zigbee mesh network using multiple Arduino boards and Zigbee modules. Configure the network with a coordinator node and several router nodes. Program the nodes to transmit data in a multi-hop manner, allowing the data to traverse through multiple nodes to reach the coordinator. Demonstrate the self-healing and self-organizing capabilities of Zigbee mesh networking by dynamically adding or removing nodes from the network.

**Experiment 9: IoT connectivity with Cloud:**

* IoT Connectivity: Connecting the ESP32 board to the internet using the ESP32 module and MQTT protocol.
* Remote Sensing with GSM: Sending any sensor data to a cloud platform and ThingSpeak.

**Experiment 10: Node Red and Security:**

* Real-Time Data Visualization: Creating a real-time data visualization dashboard using Node-RED to monitor sensor readings.
* IoT Security: Implementing security measures, such as encryption and authentication, to secure IoT communications and data.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. MIsha Dohler, "Internet of Things: Architectures, Protocols, and Standards", 2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, and Robert Barton, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, 2017. |
| **Reference books** | 1. Yan Zhang, Laurence T. Yang, and Huansheng Ning, "Wireless Communications and Networks for the Internet of Things", Auerbach Publication, 2019. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSIS3023P** | **Industrial IoT and ARM based Embedded Programming** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **IoT Network Architecture and Communication Protocols- CSIS3024P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand ARM Architecture and Embedded Programming.
2. Explore Industrial IoT (IIoT) and IIoT Applications.
3. Learn about Industrial Automation and SCADA Systems:
4. Understand Medical IoT and Healthcare Applications

**Course Outcomes**

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

**CO1.** Students will be able to develop Proficiency in ARM Architecture and Embedded Programming.

**CO2.** Students will be able to program ARM controllers for sensor interfacing and data processing in embedded C.

**CO3.** Students will be able to apply Industrial IoT Principles and Technologies.

**CO4.** Students will be able to explore IIoT Applications in Different Industries.

**CO5.** Students will be able to understand Medical IoT and Healthcare Applications:

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |  | 1 |
| **CO4** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | 2 | 2 |
| **Average** | **-** | **-** | **-** | **-** | 1.2 | **-** | **-** | **-** | .8 | **-** | **-** | **-** | **-** | 1.6 | 1.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: ARM Architecture 12 Lecture Hours**

1. ARM Architecture Overview
2. 32 bit MC benefits
3. ARM vs Intel
4. Bus and Register Structures
5. Pins and Ports
6. ARM based processor family
7. ARM Cortex M Series
8. Vector Tables

**Unit II: Programming ARM boards 12 Lecture Hours**

1. STM32 Boards based on ARM
2. GPIO Programming
3. USART Protocol
4. USART Receiver and Transmitter driver
5. Developing ADC Drivers
6. System Tick (SysTick) Timer
7. General Purpose Timers in ARM
8. Interrupt Programming for GPIO, UART and ADC
9. DMA Driver Development
10. I2C overview and driver development
11. SPI overview and driver development

**Unit III: Introduction to Industrial IoT 10 Lecture Hours**

1. Industrial Revolutions
2. Industry 4.0 revolutions
3. Cyber Physical system and next generation sensors
4. IIoT Computing
5. Digital Twin Technology
6. Digital Twin Conceptual Architecture
7. AWS Digital Twin Maker
8. Software defined networking in IIoT
9. Security in IIoT

**Unit IV: Industrial Automation and IIoT Applications 14 Lecture Hours**

1. Introduction to PLC
2. Evolution, Types, advantages, limitations and applications of PLC
3. Introduction to SCADA
4. Need and Applications
5. SCADA Architecture
6. Human Machine Interfaces (HMI)
7. Smart Home and Cities
8. Industry Overview and IIoT applications in
   1. Oil, Chemical, Pharmaceutical, Manufacturing Industry
   2. Logistic Industry
   3. Smart Agriculture
   4. Other Industries

**Unit V: Medical IoT 12 Lecture Hours**

1. Healthcare Industry and Challenges
2. Telemedicine
3. Health Vital signs and symptoms
4. Wearable sensors
5. Body Area Network
6. Sensors for Medical applications

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Alasdair Gilchrist, “Industry 4.0: The Industrial Internet of Things”, Apress, 2017. 2. Sabina Jeschke, Christian Brecher, Houbing Song, and Danda B. Rawat, “Industrial Internet of Things: Cybermanufacturing Systems”, Springer, 2018. |
| **Reference books** | 1. Muhammad A. Mazidi, Shujen Chen, Eshragh Ghaemi, “STM32 Arm Programming for Embedded Systems”, Microdigitaled, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Industrial IoT and ARM based Embedded Programming Lab**

**List of Experiments**

**Experiment 1: Introduction to ARM Programming IDE:**

* Installing STM32Cube IDE and exploring the environment
* Exploring the documentation for ARM based micro controllers
* Overview and exploration of ARM cortex M series
* Exploring the ports and pins of the Nucleo boards.

**Experiment 2: Introduction to ARM Programming:**

* LED Blinking: Write a program to control an LED connected to an ARM-based board and make it blink at regular intervals.
* Interface a push-button with the ARM board and turn ON/OFF an LED.

**Experiment 3: ARM GPIO Programming:**

* Control GPIO outputs using the bit set/reset register (BSRR)
* Develop the GPIO input driver for reading value from the potentiometer
* Utilize timers to generate PWM signals on GPIO pins. Control the intensity of an LED or a servo motor using Pulse Width Modulation (PWM).

**Experiment 4: UART Protocol:**

* Develop the UART transmitter protocol to send any random number on the LCD display
* Develop the UART receiver protocol to read temperature sensor value and display it on the LCD.

**Experiment 5: ADC Programming**

* Develop the ADC single conversion cycle to read the data from an analog sensor.

**Experiment 6: ARM SysTick timer**

* Demonstrate and write the System tick timer driver using the Nucleo board.
* Use the SysTick Timer to blink the onboard LED on the Nucelo board.

**Experiment 7: STM HAL Library**

* Implement a GPIO (General Purpose Input/Output) experiment using STM HAL library on Nucleo board. Configure GPIO pins for input and output modes, toggle LED connected to an output pin, and read status from a push-button connected to an input pin.
* Set up a UART communication experiment using STM HAL library on Nucleo board. Configure UART module with desired baud rate, and establish bi-directional communication between Nucleo board and an external device, such as a PC, to exchange data.

**Experiment 8: ARM Interrupt Timers**

* Develop the GPIO interrupt timer for blinking an external LED on pushing external push button.
* Develop and demonstrate the UART transmitter timer.

**Experiment 9: SPI Protocol Interfacing**

Develop the SPI diver to Interface the ADXL accelerometer and read the sensor values.

**Experiment 10: I2C Protocol Interfacing**

Develop the I2C diver to Interface the ADXL accelerometer and read the sensor values.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Alasdair Gilchrist, “Industry 4.0: The Industrial Internet of Things”, Apress, 2019. 2. Sabina Jeschke, Christian Brecher, Houbing Song, Danda B. Rawat, “Industrial Internet of Things: Cybermanufacturing Systems”, Springer, 2017. |
| **Reference books** | 1. Muhammad A. Mazidi, Shujen Chen, and Eshragh Ghaemi, “STM32 Arm Programming for Embedded Systems”, Microdigitaled, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSIS4015P** | **Single Board Computers and IoT Applications Development** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Industrial IoT and ARM based Embedded Programming - CSIS3023P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand Single Board Computers and Embedded System.
2. Master Raspberry Pi and Linux-based Systems.
3. Develop Skills in Programming and Interfacing with Raspberry Pi.
4. Mastering cloud technologies for IoT applications.

**Course Outcomes**

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

**CO1.** Students will be able to demonstrate Proficiency in Single Board Computers and Embedded Systems.

**CO2.** Students will apply Real-Time Operating Systems (RTOS) Principles.

**CO3.** Students will be able to utilize Raspberry Pi and Linux-Based Systems.

**CO4.** Students will be able to develop Skills in Programming and Interfacing with Raspberry Pi:

**CO5.** Students will be able to apply IoT Principles and Cloud Integration.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 |
| **CO4** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | 2 | 2 |
| **Average** | **-** | **-** | **-** | **-** | 1.2 | **-** | **-** | **-** | .8 | **-** | **-** | **-** | **-** | 1.6 | 1.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction 12 Lecture Hours**

1. System on Chip Overview
2. ARM Cortex series as SoC
3. Single Board Computers
4. Components and configurations
5. SBC Types
6. ASUS Tinkerboard
7. BeagleBone Black
8. NVIDIA Jetson
9. Latte Panda
10. Raspberry Pi

**Unit II: RTOS Basics 12 Lecture Hours**

1. RTOS Need and challenges
2. RTOS Task Scheduling
3. Real time design issues
4. Multi-threading models, threading issues, thread libraries, synchronization Mutex: creating, deleting, prioritizing mutex, mutex internals
5. Messages, Buffers, mailboxes, queues, semaphores
6. Free RTOS Overview

**Unit III: Raspberry Pi Introduction 12 Lecture Hours**

1. Linux- Introduction, Architecture, File System
2. Linux commands, sudo, apt
3. Raspbian O.S.- Introduction
4. Rasperry Pi Pin description
5. RPi configurations
6. On board components on RPi
7. SoC used for RPi
8. Pi Pico Introduction
9. Pico vs Arduino Nano
10. Programming in Pico

**Unit IV: Programming and Interfacing 12 Lecture Hours**

1. Sensors Interfacing- Temperature and Humidity Sensor (DHT11), Motion Sensor (PIR), Obstacle detection using Ultrasonic sensor
2. Communicating using RPi- GSM interfacing, Accessing on-board Wi-Fi
3. Connecting Database with RPi
4. Using Wiring Pi for GPIO Programming
5. Various sensors, modules interfacing with SPI, I2C and UART
6. Interfacing Rpi using C
7. SSH command and working
8. RPi as SSH
9. RPi as Web server
10. GPIO Control over WebBrowser
11. Node-RED, MQTT Protocol
12. Camera interfacing

**Unit V: Cloud Technologies for IoT** **12 Lecture Hours**

1. Overview of GCP, AWS, Azure, ThingSpeak
2. Overview of Azure cloud computing platform and services
3. Key Azure services, such as virtual machines, storage, networking, and databases
4. Introduction to Azure IoT Hub
5. Understanding IoT device connectivity and messaging with Azure IoT Hub
6. Provisioning IoT devices with Azure IoT Hub and managing device identities
7. Device authentication and authorization with Azure IoT Hub
8. Connecting Azure IoT Hub with Azure Stream Analytics for real-time data processing
9. Connecting RPi with Azure
10. Raspberry Pi data logging and visualization with Azure

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Simon Monk, "Programming the Raspberry Pi", 2nd Edition, McGraw Hill TAB, 2015. 2. Ritesh Modi, "Azure for Architects” by Packt Publishing Limited, 2017. |
| **Reference books** | 1. 1. Doug Abbott, “Linux for Embedded and Real-Time Applications”, Newnes, 2003. 2. Richard Barry, “Mastering the FreeRTOS™ Real Time Kernel”, Real Time Engineers Ltd. Available online at:   <https://freertos.org/Documentation/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf> |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Single Board Computers and IoT Applications Development Lab**

**List of Experiments**

**Experiment 1: Introduction to Raspberry Pi Pico:**

* To familiarize with the Raspberry Pi Pico GPIO and its programming by blinking an LED.
* Interface an analog sensor (e.g., potentiometer) with the Raspberry Pi Pico, and read analog values using the built-in ADC.
* Connect a DHT11 or DHT22 sensor to the Raspberry Pi Pico and read temperature and humidity data using GPIO pins and the sensor library.

**Experiment 2:** Real-Time Operating Systems (RTOS) on Raspberry Pi Pico**:**

* Install and configure FreeRTOS on Raspberry Pi to understand its basic functionalities and features.
* Implement task scheduling and prioritize and run multiple tasks for real-time execution using FreeRTOS.

**Experiment 3:** Raspberry Pi Introduction and Linux-Based Systems**:**

* Explore the Linux operating system and its architecture, including the file system and shell commands.
* Install, configure Raspbian OS on Raspberry Pi and familiarize with its boot process and essential services.
* Configure a buzzer and play various frequency tones.

**Experiment 4:** Sensor Interfacing with Raspberry Pi

* Interface temperature and humidity sensor (DHT11) with Raspberry Pi to collect environmental data.
* Set up a motion sensor (PIR) to detect motion and play buzzer using Raspberry Pi..
* Develop obstacle detection using an ultrasonic sensor and process the data for obstacle avoidance and play buzzer with various patterns as per obstacle distance.

**Experiment 5:** Communication Methods on Raspberry Pi

* Utilize GSM interfacing on Raspberry Pi to send and receive SMS messages or data.
* Establish Wi-Fi connectivity on Raspberry Pi to communicate with other devices or the internet.
* Using SPI protocol to connect to an appropriate sensor and display the readings.
* Using I2C protocol to connect to accelerometer sensor and display the readings.

**Experiment 6:** GPIO Programming and Control

* Develop GPIO programming using Wiring Pi library to control LEDs, buttons, and other peripherals.
* Interface a servo motor with Raspberry Pi and control its movements through GPIO pins.
* Implement PWM control on Raspberry Pi to adjust the brightness of an LED or control motor speed.

**Experiment** 7: Node-RED and MQTT Protocol Integration

* Set up Node-RED on Raspberry Pi and create flows to process and visualize sensor data.
* Implement MQTT communication protocol to enable message exchange between Raspberry Pi and other IoT devices.
* Integrate Node-RED and MQTT to create a simple IoT application for real-time data streaming.

**Experiment 8:** Raspberry Pi Camera Module

* interface the Raspberry Pi Camera Module with the Raspberry Pi, install the necessary libraries, and capture images using Python code.
* To record videos using the Raspberry Pi Camera Module, configure video settings such as resolution and frame rate, and save the recorded video to the storage.

**Experiment 9:** Microsoft Azure IoT/ AWS IoT/ GCP Hub Integration

* Connect Raspberry Pi with Microsoft Azure IoT Hub for secure device connectivity.
* Provision IoT devices on Azure and manage their identities for secure communication.
* Implement device authentication and authorization to ensure data security on the Azure platform.

**Experiment 10:** Raspberry Pi-Based Data Logging and Visualization with Azure or AWS or GCP:

* Implement data logging on Raspberry Pi to collect and store sensor data over time.
* Use Azure Stream Analytics to process and analyze real-time data from Raspberry Pi.
* Visualize the analyzed data using Azure tools like Power BI for effective data representation.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Simon Monk, "Programming the Raspberry Pi”, 2nd Edition, by McGraw Hill TAB, 2015. 2. Ritesh Modi, "Azure for Architects” by Packt Publishing Limited, 2017. |
| **Reference books** | 1. Doug Abbott, “Linux for Embedded and Real-Time Applications”, Newnes, 2003. 2. 2. Richard Barry, “Mastering the FreeRTOS™ Real Time Kernel”, Real Time Engineers Ltd. Available online at:  <https://freertos.org/Documentation/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf> |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSBA4023P** | **Data Analytics for IoT** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the Fundamentals of IoT Data Analytics:
2. Master Time Series Data Analysis Techniques for IoT:
3. Develop Machine Learning Competence for IoT Data:
4. Apply TensorFlow for IoT Data Analytics:

**Course Outcomes**

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

**CO1.** Students will be able to apply Data Analytics Techniques to IoT Data.

**CO2.** Students will be able to analyze Time Series Data for IoT Applications.

**CO3.** Students will be able to implement Machine Learning Algorithms for IoT Data.

**CO4.** Students will be able to utilize TensorFlow for IoT Data Analytics.

**CO5.** Students will be able to apply IoT Data Analytics in Real-World Use Cases.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |  | 1 |
| **CO4** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | 2 | 2 |
| **Average** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 2 | 1.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to IoT Data Analytics 8 Lecture Hours**

1. Understanding the role and significance of data analytics in IoT applications
2. Types of Data Analysis
   1. Descriptive
   2. Diagnostics
   3. Predictive
   4. Prescriptive
3. Structured vs Unstructured Data
4. Use cases for IoT Data Analytics
   1. Inventory management
   2. Predictive Maintenance
   3. Healthcare

**Unit II: Time Series Data 12 Lecture Hours**

1. Characteristics and properties of time series data
2. Types of time series data, such as univariate and multivariate
3. Applications and challenges
4. Anomaly and Outlier detection in time series
5. Autocorrelation, Seasonality, Stationarity
6. Forecasting methods ARIMA, SARIMA, Moving Average and exponential smoothing
7. Seasonality and trend analysis in time series data
8. Decomposition methods for separating trend, seasonality
9. Time series clustering and segmentation techniques
10. Spectral analysis and wavelet transforms
11. EEG Data analysis example

**Unit III: Machine Learning Basics 10 Lecture Hours**

1. Supervised vs Unsupervised learning vs Reinforcement Learning
2. Linear Regression
3. Curve Fitting and Polynomial Regression
4. Logistic Regression
5. Classification with SVM, Decision Tree, Neural Network
6. Ensemble Learning
7. Dimensionality Reduction
8. Deep Learning Overview
9. Convolution Technique
10. Activation functions
11. Convolution Neural Network
12. Other models

**Unit IV: TensorFlow for IoT**  **10 Lecture Hours**

1. Tensorflow
2. Tensor in Tensorflow
3. Tensorflow constants, variables
4. Tensorflow keras model
5. Compile a tensorflow model
6. Training and evaluating
7. TensorFlow Lite
8. Features of TensorFlow Lite
9. TensorFlow Lite on Arduino
10. TensorFlow Lite on Raspberry Pi

**Unit V: Use Case Discussion 5 Lecture Hours**

1. Problem formulation
2. Detailed solution with IoT
3. Data collection and cleaning
4. Model curation and parametrization
5. Testing and Validation

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Suhasini Subba Rao, “A course in Time Series Analysis”, 2022. Available online at: . 2. Aurélien Géron, “Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 3rd Edition, Shroff/O'Reilly, 2022 |
| **Reference books** | 1. Karthikeyan NG, "Machine Learning Projects for Mobile Applications”, Packt Publishing Limited, 2018. 2. Robert Shumway, and David Stoffer, “Time Series: A Data Analysis Approach Using R”, Chapman and Hall/CRC, 2019. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Data Analytics for IoT Lab**

**List of Experiments**

|  |  |
| --- | --- |
| **Lab. Exercise 1** | Introduction to Data Analytics for IoT |
| **Lab. Exercise 2** | Time series programming lab 1  a) Exploring time series data and its attributes  b) Working out forecasting methods. |
| **Lab. Exercise 3** | Time series programming lab 1  a) Experimenting trend analysis  b) Clustering and segmentation techniques |
| **Lab. Exercise 4** | Programming exercise 1: Exploring machine learning basics. |
| **Lab. Exercise 5** | Programming exercise 2: Exploring machine learning basics. |
| **Lab. Exercise  6** | Implementation of CNN on an available dataset. |
| **Lab. Exercise  7** | Exploring tensor flow library for IoT data analytics. |
| **Lab. Exercise  8** | Using tensor flow for training and evaluating a model |
| **Lab. Exercise  9** | Implementation of Tensorflow Lite on Arduino |

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. “A course in Time Series Analysis” by Suhasini Subba Rao 2. 2. “Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow” by Aurélien Géron |
| **Reference books** | 1. " Machine Learning Projects for Mobile Applications” by Karthikeyan NG 2. “Time Series: A Data Analysis Approach Using R” by  Robert Shumway and David Stoffer |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

**9 . Graphics & Gaming Track**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Interactive Design and 3D Animation** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 04** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Python programming - CSEG1021** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

Students will be able to learn:

1. Students will be able to learn design thinking methodologies and principles, applying them to problem-solving and user-centered design.
2. Students will be able to learn proficiency in industry-standard design software tools, effectively utilizing them in graphics, 3D modeling, animation, UI/UX design, and interactive design.
3. Students will be able to learn the application of design principles, such as typography, color theory, and gestalt principles, to create visually appealing and effective designs.
4. Students will be able to learn skills in 3D modeling, animation techniques, and interactive design, enabling them to create captivating and engaging interactive experiences.

**Course Outcomes**

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

**CO1:** Demonstrate a comprehensive understanding of design thinking methodologies and principles and apply them effectively to solve complex design problems and create user-centered interactive experiences.

**CO2:** Apply user experience (UX) design principles and conduct user research to design intuitive and user-friendly interfaces, and effectively communicate user flows and interactions through wireframes and prototypes.

**CO3:** Apply design principles, such as typography, color theory, and gestalt principles, to create visually appealing and cohesive designs that effectively communicate the intended message and engage users.

**CO4:** Create high-quality 3D models, animations, and interactive elements, demonstrating proficiency in using relevant software and techniques to bring designs to life.

**CO5:** Develop proficiency in industry-standard design software tools, utilizing them proficiently in graphics, 3D modeling, animation, UI/UX design, and interactive design projects.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | 2 |
| **CO 3** | **-** | **-** | **-** | **-** |  | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 |
| **CO4** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | 2 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | 3 |
| **Average** | **-** | **-** | **-** | **-** | 1.2 | **-** | **-** | **-** | 1.2 | **-** | **-** | **-** | **-** | **-** | 2.2 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Design Fundamentals 15 Lecture Hours**

Theory and Design Thinking, Introduction to Graphics and Visual Communication, Design Principles and Composition, Typography and Color Theory, Gestalt Principles and Advanced Design Techniques.

**Unit II: UI/UX Design 15 Lecture Hours**

User Experience (UX) Design Principles and Methodologies, User Interface (UI) Design Best Practices

Designing Intuitive and User-Friendly Interfaces, Conducting User Research and Usability Testing, Creating Interactive Elements and User Interactions, Designing User Flows and Wireframes, Prototyping Interactive Designs using FIGMA, Photoshop.

**Unit III: 3D Modelling and Animation.**  **15 Lecture Hours**

Blender/Maya Interface and Navigation, Mesh Modeling Techniques, UV Mapping and Texturing, Rigging, Lightning and Rendering. Principles of Animation by Disney, Techniques for 2d Animation and 3D Animation.

**Unit IV: Capstone Projects 15 Lecture Hours** Project Planning and Concept Development, Pre-production: research, concept art, storyboards, script

Production: asset creation, character rigging, animation, post-production: refining animation, adding sound effects and music, Presentation and Documentation: final presentation, project documentation, portfolio creation.

A capstone project depicting the above skillsets in the following project categories. Short Film Production, Character Animation Showcase, Motion Graphics Advertisement, Game Animation, Visual Effect Showcase.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Jennifer Preece, Yvonne Rogers, and Helen Sharp “Interaction Design: Beyond Human-Computer Interaction", 5th Edition, Wiley, 2019. 2. Todd Palamar, "Mastering Autodesk Maya 2022: A Comprehensive Guide to 3D Animation and Modeling". |
| **Reference books** | 1. Ben Simonds, “Blender Master Class: A Hands-On Guide to Modeling, Sculpting, Materials, and Rendering", No Starch Press, 2013. 2. Don Norman, “The Design of Everyday things”, Hachette Audio, 2018. 3. Richard E Williams, “The Animator’s Survivor Kit”, Faber & Faber, 2009. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Introduction to Interactive Design and 3D Animation Lab**

**List of Experiments**

**Experiment 1: Introduction to Graphic Design Principles and Composition**

1. Create a mood board using visual elements to evoke specific moods or styles.
2. Design a simple composition applying graphic design principles.

**Experiment 2: Typography and Color Theory Exploration**

1. Design a typographic poster using appropriate typefaces, font sizes, and hierarchy.
2. Apply color theory principles to create a visually appealing color scheme.

**Experiment 3: Gestalt Principles and Advanced Design Techniques**

1. Create a series of design compositions demonstrating the application of Gestalt principles.
2. Experiment with advanced design techniques such as visual illusions, negative space, or unconventional layouts.

**Experiment 4: User Experience (UX) Design Principles, Usability Studies**

1. Conduct a small-scale user research study, including interviews or surveys.
2. Analyze the collected data and create user personas.
3. Present findings and recommendations for improving the user experience of a selected application.

**Experiment 5: Designing User Flows and Wireframes**

1. Design user flows illustrating the step-by-step process and navigation of a selected application or website.
2. Create wireframes for key screens or pages considering information hierarchy and user interactions.
3. Present wireframes, explaining the design decisions and justifications.

**Experiment 6: Prototyping Interactive Designs using Figma**

1. Create interactive prototypes using Figma, incorporating user interactions, transitions, and feedback elements.
2. Conduct usability testing on the prototypes and gather feedback for refinement.

**Experiment 7: Introduction to 3D Modelling**

1. Use 3D modeling software to create simple 3D models of objects.
2. Apply learned techniques to add realism to the models through texture mapping.

**Experiment 8: Advance 3D Modelling**

1. Apply UV mapping techniques to unwrap 3D models.
2. Add textures to the models, considering lighting and shading principles.
3. Render the models to create visually appealing 3D scenes.

**Experiment 9: Animation**

1. Create a short 2D animation applying animation principles and techniques.
2. Explore basic 3D animation techniques, including keyframing, rigging, and character animation.

**Experiment 10: Capstone Project**

1. Work on a comprehensive design project integrating graphic design, UX design, and 3D modeling or animation elements.
2. Create a final deliverable, such as a presentation or prototype, showcasing the project's design process and outcomes.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Jennifer Preece, Yvonne Rogers, and Helen Sharp “Interaction Design: Beyond Human-Computer Interaction", 5th Edition, Wiley, 2019.  2. Todd Palamar, "Mastering Autodesk Maya 2022: A Comprehensive Guide to 3D Animation and Modeling". |
| **Reference books** | 1. Ben Simonds, “Blender Master Class: A Hands-On Guide to Modeling, Sculpting, Materials, and Rendering", No Starch Press, 2013. 2. Don Norman, “The Design of Everyday things”, 2nd Edition, Basic Books, 2013. 3. Richard E Williams, “The Animator’s Survivor Kit”, Faber & Faber, 2009. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSGG3024P** | **Game Programming** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Interactive Design and 3D Animation -**  **CSGG2012P** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

Students will be able to:

1. Understand the game design process, including game genres, storytelling, and prototyping.
2. Develop programming skills for game development, covering variables, control structures, OOP, scripting, and debugging.
3. Gain proficiency in Unity game engine for scene management, asset manipulation, physics, AI, UI, and optimization.
4. Explore advanced game development topics, such as procedural content generation, advanced AI, audio, and visual effects.
5. Apply knowledge through a capstone project, encompassing project planning, development, iteration, and presentation.

**Course Outcomes**

Students will be able to:

1. Analyze and evaluate game design principles and concepts, demonstrating an understanding of the game development process, game genres, and storytelling techniques.
2. Apply programming fundamentals for game development, utilizing variables, control structures, object-oriented programming principles, scripting languages, and debugging techniques.
3. Demonstrate proficiency in using the Unity game engine, including project setup, scene management, asset manipulation, physics simulation, user input and controls, AI implementation, user interface design, sound design, and visual effects.
4. Explore advanced game development topics, such as procedural content generation, advanced physics simulation, advanced AI techniques, advanced user interface design, advanced audio design, and advanced visual effects and shaders.
5. Plan, manage, and develop a game project using the knowledge and skills acquired throughout the course, iterating on the project based on feedback, and presenting the final game project to showcase competence in game development.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 2** | 1 | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | **-** |
| **CO 3** | 1 | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | 1 | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **Average** | 1 | **-** | 1.6 | **-** | 1.2 | **-** | **-** | **-** | 0.8 | **-** | **-** | **-** | 1.2 | 0.4 | 1.8 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Game Design and Development 12 Lecture Hours**

Overview of game design and development process, Game Development Life Cycle, Introduction to game engines and their importance, Understanding the roles in game development teams, Fundamentals of game design principles and concepts, Introduction to game genres and platforms, Game ideation and concept development, Game storytelling and narrative design, Game prototyping and playtesting, Game Design Document

**Unit II: Programming Fundamentals for Game Development 12 Lecture Hours**

Introduction to programming languages used in game development, Variables, data types, and operators, Control structures (conditionals and loops), Functions and procedures, Object-Oriented Programming (OOP) principles for game development, Event-driven programming in game development, Memory management in game programming, Debugging and error handling in game development, Scripting languages for game development, Scripting for game mechanics and interactions, Programming patterns in game development(Singleton Pattern, Factory Pattern etc.)

**Unit III: Game Development with Unity 12 Lecture Hours**

Introduction to Unity game engine and its features, Unity interface and project setup, Working with scenes and game objects, Introduction to components and scripts in Unity, Scripting in Unity using C#, Importing and manipulating game assets (e.g., models, textures, audio), Physics simulation and collisions in Unity, User input and controls in Unity, AI and pathfinding in game development with Unity, User interface design and implementation in Unity, Sound design and implementation in Unity, Visual effects and shaders in Unity, Game optimization and performance in Unity, Networking and multiplayer implementation in Unity, Ray-casting, Tilemaps, Animator.

**Unit IV: Advanced Game Development Topics 12 Lecture Hours**

Procedural content generation in game development, Advanced physics simulation in game development, Advanced AI and behavior trees in game development, Advanced user interface design and implementation, Advanced audio design and implementation, Advanced visual effects and shaders, Game analytics and player behavior analysis, Cross-platform game development, Emerging technologies in game development

**Unit V: Capstone Project: Game Development 12 Lecture Hours**

Game project planning and management, Game project development and iteration, Game project polishing and finalization, Game project presentations and feedback.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Harrison Ferrone, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018. 2. Robert Nystrom, “Game Programming Patterns”, The Pragmatic Bookshelf, 2014. 3. Jeremy Gibson Bond, “Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#”, 3rd Edition, Addison-Wesley Professional, 2022. 4. Joe Hocking, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018. |
| **Reference books** | 1. Alan Thorn, “Mastering Unity 2D Game Development”, Packt Publishing, 2014. 2. Harrison Ferrone, “Learning C# by Developing Games with Unity”, 7th Edition, Packt Publishing, 2022. 3. Simon Jackson, “Mastering Unity 2D Game Development”, Packt Publishing, 2014. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Game Programming Lab**

**List of Experiments**

1. **Generating a Graphical Design Document (GDD):**
   * Define the game concept and objectives.
   * Identify the target audience and their preferences.
   * Create a visual representation of game characters, environments, and user interfaces.
   * Outline the gameplay mechanics and rules.
   * Specify the audio and visual elements.
   * Document the technical requirements and constraints.
2. **Exploring the Fundamentals of C#: Variables, Control Statements, and Loops:**
   * Understand the concept of variables and their types.
   * Learn how to declare and initialize variables in C#.
   * Explore control statements such as if-else, switch, and loops like for, while, and do-while.
   * Practice using conditional statements and loops in simple coding exercises.
   * Understand the importance of variable scope and its impact on program execution.
3. **Mastering Object-Oriented Programming using C#:**
   * Understand the principles of object-oriented programming (OOP).
   * Learn about classes, objects, and their relationships.
   * Explore encapsulation, inheritance, and polymorphism concepts.
   * Practice creating classes, defining properties and methods.
   * Implement inheritance and understand its benefits in code reuse and organization.
4. **Setting up Unity: Installation, Utilizing Unity Hub, and Acquiring Licenses:**
   * Download and install the Unity game engine.
   * Set up Unity Hub to manage projects and versions.
   * Create a Unity ID and acquire necessary licenses (if applicable).
   * Familiarize yourself with the Unity Editor's interface and features.
   * Configure Unity preferences and project settings.
5. **Immersing in Unity3D: Creating a Two-Player Coin Collection Game:**
   * Create a new Unity project and set up a 3D environment.
   * Design player characters and coin model.
   * Implement player movement and collision detection.
   * Program the coin collection mechanics and keep track of player scores.
   * Set win/lose conditions and implement game over functionality.
6. **Enhancing User Experience: Working with Unity3D UI Elements and Implementing Game Start and Closure Functions:**
   * Add UI elements like buttons, text boxes, and menus to the game interface.
   * Implement a start menu to allow players to begin the game.
   * Create a game over screen with options for replaying or exiting the game.
   * Add sound effects and visual cues to enhance the user experience.
   * Implement functionality to pause or exit the game during gameplay.
7. **Advanced Capabilities of the Unity Game Engine: Creating a Mouse-Controlled Star Destroyer Game:**
   * Design a 2D or 3D environment for the game.
   * Implement a star destroyer character that follows the mouse cursor.
   * Define the game objective, such as collecting power-ups or avoiding obstacles.
   * Implement enemy entities or obstacles that interact with the star destroyer.
   * Program game mechanics and win/lose conditions.
8. **Multi-Platform Development: Designing a Candy Crush-style Game for Various Operating Systems:**
   * Plan the game mechanics and rules similar to Candy Crush.
   * Design the game board with colorful candies and match patterns.
   * Implement swipe or click-based controls for selecting and swapping candies.
   * Program matching algorithms to detect and remove matched candies.
   * Test and optimize the game for different operating systems, such as Windows, macOS, iOS, and Android.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Harrison Ferrone, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018. 2. Robert Nystrom, “Game Programming Patterns,” Lightning Source Inc, 2014. 3. Jeremy Gibson Bond, “Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#”, 3rd Edition, Addison-Wesley Professional, 2022. 4. Joe Hocking, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018. |
| **Reference books** | 1. Alan Thorn, “Mastering Unity 2D Game Development”, Packt Publishing. 2. Harrison Ferrone, “Learning C# by Developing Games with Unity”, 6th Edition, Packt Publishing, 2021. 3. Simon Jackson, “Mastering Unity 2D Game Development”, Packt Publishing, 2014. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSGG3023P** | **Computer Graphics** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Programming in C- CSEG1041** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

Students will be able to:

1. Students will be able to create 2D and 3D objects.
2. Students will be able to apply various transformations on the 2D and 3D objects.
3. Students will apply hidden surface removal techniques along with various shading algorithms.
4. Students will create 3D graphics with realistic effects.

**Course Outcomes**

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

**CO1.** Classify various graphics hardware and software devices.

**CO2.** Use primitive operations to create 2D and 3D objects and perform various operations thereon.

**CO3.** Perform complex 2D and 3D transformations on objects.

**CO4.** Implement various hidden surface removal techniques.

**CO5.** Create 3D realistic imagery by applying shading and coloring techniques on objects.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | 1 | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** |
| **CO 2** | 2 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | **-** | **-** |
| **CO 3** | 2 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 |
| **CO4** | 2 | 1 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 |
| **CO5** | 2 | 2 | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | 2 |
| **Average** | 1.8 | 1.2 | 1.25 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | 1.33 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: 12 Lecture Hours**

Introduction to Computer Graphics and OpenGL: Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Introduction to pixel. Why OpenGL, features in OpenGL, OpenGL operations, Abstractions in OpenGL – GL, GLU & GLUT, 3D viewing pipeline, viewing matrix specifications, a few examples and demos of OpenGL programs.

**Unit II: 12 Lecture Hours**

Scan conversion – lines, circles and Ellipses; Filling polygons and clipping algorithms

Scan Converting Lines, Mid-point criteria, Aliasing and Antialiasing, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons Clipping Lines algorithms– Cyrus Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.

**Unit III: 12 Lecture Hours**

2-D and 3-D Transformations: Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The UNIT Square, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations. Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations.

**Unit IV: 12 Lecture Hours**

Rendering : Visible-Surface Determination, Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, the Z-Buffer Algorithm, Scan-line method, Painter’s algorithms (depth sorting), Octrees , BSP trees, Visible-Surface Ray Tracing, comparison of the methods.

Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Phong’s model, Gouraud shading, some examples.

**Unit V: 12 Lecture Hours**

Plane Curves and Surfaces

Curve Representation, Nonparametric Curves, Cubic Splines, Bezier Curves, Bspline Curves, B-spline Curve Fit, Knot Vectors, NURBS, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces, Fractals.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Donald D. Hearn and M. Pauline Baker, “Computer Graphics using OpenGL”, Pearson Education India, 2014. |
| **Reference books** | 1. David F. Rogers and J. Alan Adams, “Mathematical Elements for Computer Graphics”, 2nd Edition, McGraw- Hill, Inc, 2017. 2. James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes "Computer Graphics: Principles and Practice", Pearson, 2002. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Computer Graphics Lab**

**List of Experiments**

**Experiment 1: Introduction to OpenGL: [ Lab Environment Setup]**

* What is OpenGL?
* What is GLU/GLUT?
* What is OpenGL Architecture?
* Setting up the environment.
* First OpenGL Program: This initializes a window of Green color.
* Draw a Hut.

**# Discuss all the steps & functions in the program.**

**Experiment 2: Event Handling**

* Implement mouse input functionality.
* Implement keypress functionality.
* Implement another call back functions.

**#Implement above with the help of animation.**

**Imp: For all the remaining experiments, take the input with the help of mouse clicks, don’t use console for any type of input.**

**Experiment 3: Drawing a line [Usage of Open GL on Linux Environment for Virtual Environment]**

* Draw a line using equation of line Y=m\*X+C.
* Draw a line using DDA algorithm for slope m<1 and m>1.
* Draw a line using Bresenham algorithm for slope m<1 and m>1.

**# Take the input from user for all the three scenarios i.e. value of (x1, y1) and (x2, y2).**

**Experiment 4: Drawing a Circle and an Ellipse [Done on OpenGL Environment]**

* Draw the circle with the help of polar equations
* Draw the circle with the help of mid-point method.
* Draw the Ellipse with the mid-point method.

**# Take the value of radius, major axis and minor axis as input from the user.**

**Experiment 5 Seed Fill Algorithms [Small Project will be given for demonstration]**

* WAP to fill the polygon using scan lines.
* WAP to fill a region using boundary fill algorithm using 4 or 8 connected approaches.
* WAP to fill a region using flood fill algorithm using 4 or 8 connected approaches.

**# Take the value of seed point, intensity of new color as input from user.**

**Experiment 6: Viewing and Clipping [Geographical Animation for demonstration]**

* Write an interactive program for line clipping using Cohen Sutherland line clipping algorithm.
* Write an interactive program for line clipping using Liang-Barsky line clipping algorithm.
* Write an interactive program for polygon clipping using Sutherland – Hodgeman polygon clipping algorithm.

**# Take the window coordinates as input from the user, also take polygon coordinates as input.**

**Experiment 7 : Basic 2D & 3D Transformations**

* Write an interactive program for following basic transformation.
* Translation
* Rotation
* Scaling
* Reflection about axis.
* Reflection about a line Y=mX+c and aX+bY+c=0.
* Shear about an edge and about a vertex.

**# Perform all the experiment for 3-D transformation.**

**# Take the following values as input from user: Theta (angle of rotation), translation factor, scaling factor and other values. Make necessary assumptions.**

**Experiment 8: Drawing Bezier curves. [ Virtual GLUT based demonstration]**

* Write a program to draw a cubic spline.
* WAP to draw a Bezier curve**.**

**# Take necessary values as input from the user like degree of the Bezier curve.**

**Experiment 9: VSD and Shading Algorithms.**

* Implement VSD Algorithms like, Backface, Z Buffer.
* Implement Raycasting algorithm.

**Experiment 10: Illumination and shading Algorithms.**

* Implement Phong illumination model for computing light intensity
* Implement Shading algorithms: Flat, Phong and Gourard Shading.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. 1. Donald D. Hearn and M. Pauline Baker, “Computer Graphics using OpenGL”, Pearson Education India, 2014. |
| **Reference books** | 1. 1. Dave Shreiner, John Kessenich, Bill Licea-Kane, The Khronos OpenGL ARB Working Group, “OpenGL: Programming Guide, the Official Guide to Learning OpenGL”, 8th Edition, Addison Wesley, 2013. 2. 2. Mason Woo, and Dave Shreiner, “OpenGL Programming Guide”, Paperback, 2008. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSGG4015P** | **Augmented and Virtual Reality Development** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the fundamental concepts and principles of Augmented Reality (AR) and Virtual Reality (VR), including their differences, applications, and potential impact on various industries.
2. Gain proficiency in using AR/VR hardware and software tools, including AR/VR devices, headsets, controllers, and development platforms.
3. Develop practical skills in creating AR/VR experiences by learning the necessary techniques and technologies, such as marker-based and marker less AR tracking, VR environment design, and user interaction implementation.
4. Acquire hands-on experience in AR/VR development using the Unity game engine, including asset importing, scene creation, scripting, and performance optimization.
5. Apply user experience (UX) and interaction design principles to create immersive and user-friendly AR/VR interfaces, and understand the challenges and considerations involved in designing and testing AR/VR applications.

**Course Outcomes**

**CO1**: Students will demonstrate knowledge and understanding of AR/VR concepts, technologies, and applications.

**CO2:** Students will apply their knowledge to create AR/VR experiences using industry-standard tools.

**CO3:** Students will analyze the effectiveness of different AR/VR design choices and techniques. They will assess the impact of these decisions on user experience, performance, and overall quality of AR/VR applications.

**CO4:** Students will critically evaluate AR/VR applications in terms of usability, user experience, and effectiveness.

**CO5:** Students will synthesize their knowledge and skills to design and develop original AR/VR applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | **-** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | **-** | 1 |
| **CO 2** | 1 | **-** | 1 | **-** | 3 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **-** | **-** | 1 |
| **CO 3** | 1 | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO4** | 1 | **-** | 2 | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | **-** | 3 |
| **CO5** | 1 | **-** | 2 | **-** | 3 | **-** | **-** | **-** | 2 | **-** | **-** | 1 | 1 | **-** | 3 |
| **Average** | 1 | **-** | 1.8 | **-** | 2.2 | **-** | **-** | **-** | 2 | **-** | **-** | 1 | 1.5 | **-** | 2.2 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Augmented and Virtual Reality 10 Lecture Hours**

Introduction to AR/VR, history of AR/VR, definition of AR/VR, difference between AR/VR, applications of AR/VR, introduction to XR (Extended Reality), future trends of AR/VR, Applications: Metaverse.

**Unit II: Tools and Technologies for AR VR Development 10 Lecture Hours**

VR/MR Head mounted Display (Wired/Wireless): Oculus Quest 2, HTC Vive Pro, Oculus Quest Pro, Apple Mixed Reality. Software’s: SDKs for AR development, Game Engines,3D Modelling Software, 3D Tracking and Sensors: SLAM (Simultaneous Localization and Mapping), Audio and Spatial Sound Design.

**Unit III: AR/VR Development with Unity 24 Lecture Hours**

Introduction to Unity game engine and its AR/VR capabilities, importing assets and creating scenes for AR/VR, Scripting interactions and behaviors in Unity, Marker based AR Development using Vuforia, Marker less AR development using ARfoundation: ARCore/Arkit. Deploying Application on Playstore.

Introduction to Unity XR, VR Interaction Mechanics, VR Locomotion and Navigation, VR User Interface (UI) and User Experience (UX), VR Audio and Spatial Sound. XR Interaction Toolkit: XR Input and Controllers, Performance Optimization for VR, Testing and Deployment of VR Applications.

**Unit IV: Designing for Immersive Applications 6 Lecture Hours**

Interaction Design for Immersive Experiences, Spatial UI Design, Visual Design for Immersive Applications, Navigation and Wayfinding in Immersive Environments, Comfort and Accessibility in Immersive Design, Usability Testing, and Iterative Design for AR/VR Development.

**Unit V: Capstone Project 10 Lecture Hours**

Project proposal and planning, research and analysis, design and development, testing and iteration, documentation and presentation.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Dieter Schmalstieg and Tobias Hollerer, "Augmented Reality: Principles and Practice", Pearson, 2016. 2. Steve Aukstakalnis, "Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR", Addison-Wesley, 2016. |
| **Reference books** | 1. Tony Parisi, “Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile", O'Reilly Media, 2015. 2. Ralf Doerner, Wolfgang Broll, Paul Grimm, Bernhard Jung, “Virtual and Augmented Reality (VR/AR): Foundations and Methods of Extended Realities (XR)”, Springer, 2022. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Augmented and Virtual Reality Development Lab**

**List of Experiments**

**Experiment 1: Introduction to Augmented Reality/Virtual Reality/Mixed Reality**

1. Install Unity, the development environment for AR/VR/MR.
2. Familiarize yourself with the Unity user interface, components, and editor.
3. Learn the basics of writing scripts in Unity for creating interactive AR/VR/MR experiences.

**Experiment 2: Marker-based AR using Vuforia.**

1. Install and set up the Vuforia SDK, a plugin for AR development.
2. Create a tracker using Vuforia for marker-based AR.
3. Learn how to detect the tracker and understand the working of AR.
4. Incorporate 3D game objects into Unity scenes for AR applications.

**Experiment 3: Advanced Marker-based AR: Multiple Markers**

1. Use the Vuforia AR Camera for image-related operations.
2. Explore Multi-Image Targets in Vuforia to detect multiple images simultaneously.
3. Convert any object into an Image Target for AR tracking.
4. Play holographic videos within an augmented environment.
5. Understand Simultaneous Localization and Mapping (SLAM) in Vuforia.
6. Create virtual buttons in Vuforia and explore their applications.

**Experiment 4: Create an AR application using AR Foundation**

1. Understand the AR Foundation, a cross-platform framework for AR applications.
2. Create a basic AR application using AR Foundation, including adding 3D models, interactivity, and user interface elements.
3. Imple

**Experiment 5: Advance AR Foundation**

1. Implement light estimation in AR application.
2. Implement distance estimation in AR application.
3. Implement portal functionality in AR application.
4. Implement direct and indirect manipulation.

**Experiment 6**

1. Pick a use case and implement it either using Marker based or Marker less concept.

**Experiment 7: Introduction to VR**

1. Learn basics of VR development using Unity
2. Setting up XR Plugin for VR development.
3. Creating a simple VR environment.
4. UI Design for VR Application

**Experiment 8: VR Interaction Techniques**

1. Implementing VR interaction using controllers
2. Introducing locomotion techniques in VR
3. Implement UI interactions.
4. Incorporating realistic physics and collisions in VR
5. Implementing immersive audio and haptic feedback in VR

**Experiment 9&10: Create a VR game incorporating all the functionalities, Build the game for all platforms, Build game for all VR devices.**

1. Design and develop a VR game, including gameplay mechanics, objectives, and immersive elements.
2. Implement player movement, interaction mechanics, and game rules to create a compelling gameplay experience.
3. Optimize the game's performance and compatibility for various VR platforms and devices.
4. Test, debug, and iterate on the game to ensure smooth gameplay and fix any issues or bugs encountered.
5. Build and deploy the VR game for multiple platforms and VR devices, maximizing accessibility and user reach.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Dieter Schmalstieg and Tobias Hollerer, "Augmented Reality: Principles and Practice", Pearson, 2016. 2. Steve Aukstakalnis, "Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR", Addison-Wesley, 2016. |
| **Reference books** | 1. Jonathan Linowes and Krystian Babilinski, “Augmented Reality for Developers: Build Exciting AR Applications Using Unity, ARKit, ARCore, and Vuforia", Packt Publishing, 2017. 2. Jonathan Linowes, "Unity Virtual Reality Projects", 2nd Edition, Packt Publishing, 2018. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
| **CSGG4016P** | **Web Programming for Interactive 3D Graphics** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Understand the core principles of web programming.
2. Dive into the low-level details of WebGL, including the graphics pipeline, shader programming.
3. Develop proficiency in using the Three.js library to create complex 3D scenes, manage geometries, materials, and shaders.
4. Gain expertise in optimizing WebGL applications for performance in live Web applications.

**Course Outcomes**

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

**CO1.** Students will be able to apply HTML5, CSS3, and JavaScript to create interactive web applications.

**CO2.** Students will be able to design and implement complex 3D scenes using the WebGL API and the Three.js library.

**CO3.** Students will be able to utilize shader programming to implement advanced shading algorithms, such as Phong shading, texture mapping, normal mapping, and physically based rendering (PBR)

**CO4.** Students will be able to incorporate user interaction in 3D environments by handling input events and implementing object picking.

**CO5.** Students will be able to integrate web-based 3D graphics with other technologies, such as WebXR, to create immersive experiences and explore the possibilities of virtual reality (VR) and augmented reality (AR) in web applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 1 | 2 |
| **CO 2** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 1 | **-** | **-** | **-** | **-** | 3 | 2 |
| **CO 3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 |
| **CO4** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 1 |
| **CO5** | **-** | **-** | **-** | **-** | 2 | **-** | **-** | **-** | 2 | **-** | **-** | **-** | **-** | 2 | 2 |
| **Average** | **-** | **-** | **-** | **-** | 1.2 | **-** | **-** | **-** | 0.8 | **-** | **-** | **-** | **-** | 1.6 | 1.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Front end development for Web 10 Lecture Hours**

Introduction to the course and the History of Web Development, HTML Tags, attributes and forms

Page styling with CSS, JavaScript Basics, if-else, loop, function, DOM manipulation with JavaScript, Responsive Web UI, Bootstrap Introduction with classes, grids, markers, buttons, Bootstrap Components

**Unit II: Server-Side Programming 8 Lecture Hours**

Client Side vs Server side programming, Node JS Introduction, features and Installation, Node.js Modules, NPM and file system module, Asynchronous Programming with Node.js: Understanding the event loop and non-blocking I/O, Callbacks and handling asynchronous operations, Promises and async/await for asynchronous flow control, Express.js, Routing, Serving HTML, Images, CSS and static assets, Dynamic pages and templating, Working with Databases (MongoDB), Inserting, updating and deleting documents, REST API with Mongoose, Real-Time Communication with Socket.IO :Introduction to WebSocket communication, Building real-time applications with Socket.IO, Handling events and broadcasting messages

**Unit III: WebGL 6 Lecture Hours**

Overview of WebGL and GPU based rendering, WebGL Rendering Pipeline, WebGL Data flow concepts and major steps in programming, Applying basic and composite Transformations in WebGL, Animation loop, Adding color to a Geometry, Adding Texture to a Geometry

**Unit IV: Advance WebGL 6 Lecture Hours**

3D Programming Basics, Modeling the real world, Flow of concepts in WebGL, LookAt() method and its working, Transformation and Animation, Adding Lights, Fogs and Shadows.

**Unit V: Frameworks on WebGL 15 Lecture Hours**

Type Script Installation and Setting up development environment, THREE JS basics: Scene, Camera, Renderer and Animation Loop, Object3D and Object3D Hierarchy, Basic Geometries in THREE JS, Materials:- MeshBasic, MeshPhong, MeshNormal, MeshLambert, MeshPhysical

Texture mapping in THREE JS, Lights:- Ambient, Directional, Point, Spot Loading External Objects with GLTF, GLTF Animations loadings, Phyiscs JS, Tween JS, Raycasting and Collision Detection, WebXR overview with THREE JS, Acessign gamepad and handling controller events

WebXR API: XRSystem, XRInput, Conclusion of the course with WebGPU

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Bhupendra Singh, “Learning 3D Programming on the Web with WebGL” by 2. Jos Dirksen, "Learning Three.js: Virtual and Augmented Reality with Three.js and WebXR" by 3. Ethan Brown, "Web Development with Node and Express: Leveraging the JavaScript Stack" by |
| **Reference books** | 1. Kouichi Matsuda and Rodger Lea, "WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL", Addison-Wesley, 2013. 2. Jos Dirksen, "Learning Three.js: The JavaScript 3D Library for WebGL", Packt Pub Ltd, 2015. 3. Tony Parisi, "WebXR: Building Immersive AR and VR Applications on the Web". |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Web Programming for Interactive 3D Graphics Lab**

**List of Experiments**

**EXPERIMENT 1**

**TITLE:** **Introduction to HTML5**

**Objective: Introduction to HTML5 and CSS**

1. Create a static webpage with information about your personnel details.
2. Modify the web page created in 1, adding HTML5 features like nav, footer, header tags.
3. Modify the web page created in 2, separating all the styling related information in a separate CSS file.
4. Modify the above web page created in 3, adding video and audio files.

**EXPERIMENT 2**

**TITLE: HTML5 Forms**

**Objective: Creating a working website**

1. Create a sign-up registration page, asking the necessary details from the user. Consider the case of a college website and the students as the users.
2. For the page created in 1, add a login page and a JavaScript validating the required user’s login.
3. For the page created in 2, the user should go to personnel info. Page created in 1.4. Also add new features like session timeout using HTML5 web storage feature

**EXPERIMENT 3**

**TITLE:** **Introduction to Node JS**

**Objective :- Understanding the basic and file operations in Node.**

1. Create a simple web server in Node JS and write “Hello World” on the console.
2. Create a Node.js program that reads the contents of a text file ("sample.txt") and displays its contents to the console.
3. Build a Node.js script that accepts user input and writes it to a new text file
4. Build a Node.js script that copies the contents of one text file ("source.txt") to another file ("destination.txt"). Ensure that the destination file is created if it doesn't exist, and overwrite its contents if it does.

**EXPERIMENT 4**

**TITLE:** **RESP API**

**Objective :- Understanding the basic and file operations in Node.**

1. Build a Node.js REST API that allows users to create and retrieve resources. Resources could be simple objects with properties like "id," "name," and "description." Implement endpoints for creating new resources (POST) and retrieving a list of all resources (GET).
2. Extend the previous exercise to include the ability to update (PUT) and delete (DELETE) resources. Implement endpoints to update an existing resource by ID and delete a resource by ID.
3. Enhance the API to support pagination and filtering of resources. Implement query parameters that allow users to specify the number of results per page and filter resources based on certain criteria (e.g., filtering by name or description).
4. Secure your API by adding authentication and authorization. Users should be required to authenticate before creating, updating, or deleting resources. Implement a basic user authentication system using tokens (e.g., JWT) and restrict access to certain routes based on user roles.

**EXPERIMENT 5**

**TITLE:** **Socket programming**

**Objective :- Understanding the basic of socket programming in Node.**

1. Create a basic chat application using Node.js and the socket.io library. Students should build a real-time chat server and client where multiple users can connect, send messages, and receive messages from others in real-time.
2. Build a Node.js server that sends real-time notifications to connected clients. For example, simulate a notification system where users receive notifications whenever a certain event occurs on the server.
3. Create a collaborative drawing application where multiple users can draw together on a shared canvas in real-time. Each user's drawing actions should be synchronized with others in the room.
4. Design a simple online multiplayer game (e.g., a multiplayer tic-tac-toe or a quiz game). Students should implement game logic, real-time player interactions, and score tracking using Node.js and sockets.

**EXPERIMENT 6**

**TITLE:** **Introduction to WebGL**

**Objective :- Understanding the basics of WebGL**

1. Write a simple WebGL code displaying circle, triangle and rectangle
2. Modify program 1 and add user inputs for creating different size of circle, triangle and rectangle
3. Write a simple WebGL code allowing user to create a freehand polygon
4. Modify code of 3 and add the options of changing pencil color, size, background and choosing pre-defined shapes like circle, rectangle and ellipse

**EXPERIMENT 7**

**TITLE:** **Transformation and Viewing in WebGL**

**Objective :- Able to draw and animate 3-D objects and add realistic effects**

1. Write a WebGL program to display a rotating cuboid coloring every surface with different colors

2. For program of 1, add the control of rotation, scaling and translation to the user

3. For program of 2, add the texture to any surface of the cuboid with user defined images stored in

local repository

1. Write a WebGL program displaying ‘n’ number of bouncing balls with ‘n’ given by the user. The

program should include 3 D projection and hidden surface removal effects.

**EXPERIMENT 8**

**TITLE:** **THREE JS Introduction**

**Objective :- Able to display shapes, rotate them and apply texture.**

1. Write a THREE JS program to display any five primitive shapes available in the library.
2. Write a THREE JS program to display a rotating cuboid with rotation speed depending upon the input by the user through the web interface.
3. Load an external 3D model (e.g., a .gltf or .obj file) which is selected through web based UI into a Three.js scene and display it.
4. With the help of texture apply any skybox images over a cube.

**EXPERIMENT 9**

**TITLE:** **THREE JS Materials and lights**

**Objective :- Able to use advance materials and lights**

1. Write a THREE JS program to display 3d Models of various types materials.
2. Add various types of lights to your scene.
3. Move and rotate any 3D object in the scene using keypads and mouse.

**EXPERIMENT 10**

**TITLE:** **Advance topics in THREE JS**

**Objective :- Able to apply physics and other advanced topics.**

1. Write a THREE JS program to demonstrate a bouncing ball using PHYSICS.js
2. Use ray casting to select objects from the scene using mouse buttons.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Bhupendra Singh, “Learning 3D Programming on the Web with WebGL”. 2. Jos Dirksen, "Learning Three.js: Virtual and Augmented Reality with Three.js and WebXR", 3. Ethan Brown, "Web Development with Node and Express: Leveraging the JavaScript Stack", 2nd Edition, Shroff/O'Reilly, 2019. |
| **Reference books** | 1. Kouichi Matsuda, and Rodger Lea, "WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL", Pearson, 2013. 2. Jos Dirksen, "Learning Three.js: The JavaScript 3D Library for WebGL", Packt Pub Ltd, 2015. 3. Tony Parisi, "WebXR: Building Immersive AR and VR Applications on the Web". |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

1. **Core Computer Science**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Application of Machine Learning in Industries** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 10** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Python Programming and Applied Machine Learning** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the scope of machine learning in various public and private service industries.
2. Identify and analyze industry-specific challenges (ethical concerns, regulatory challenges) where ML can be applied.
3. To understand model interpretability and deployment.

**Course Outcomes**

After completion of course, students would be able to:

**CO 1:** Explain the need and necessity of machine learning for industrial use cases

**CO 2:** Explain key concepts of predictive maintenance and quality control in industrial settings

**CO 3:** Deploy ML models using cloud computing solutions.

**CO 4:** Analyze ethical concerns, regulatory challenges, and emerging AI trends in industries.

**CO 5:** Develop machine learning models that can be used in industrial automation and decision making.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO 11** | **PO 12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **1** | **3** | **1** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **2** |
| **CO 2** | **2** | **1** | **3** | **1** | **1** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **1** | **2** | **2** |
| **CO 3** | **2** | **-** | **2** | **1** | **1** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **2** | **1** |
| **CO 4** | **1** | **1** | **2** | **-** | **-** | **-** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **1** | **3** |
| **CO 5** | **1** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **3** | **2** |
| **Average** | **1.6** | **1** | **2.4** | **1** | **1** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **1.2** | **2** | **2** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit 1. Introduction to Machine Learning in Industries 12 Lecture Hours**

Overview of industrial revolutions (1.0 to 5.0), Introduction to machine learning, Supervised, Unsupervised and reinforcement learning, deep learning and neural networks, role of machine learning in industry automation and decision making, machine learning in industry 4.0 (automation, IoT, and efficiency) and machine learning in industry 5.0 (human centric AI, sustainability and advance AI applications)

**Unit 2. Machine Learning in Healthcare, Finance, and Cybersecurity**

**15 Lecture Hours**

ML in healthcare: Overview of AI and ML in healthcare, Importance of data driven decision making. Types of healthcare data, ML models for diagnosis and prediction, Privacy and security in healthcare data (HIPPA, GDPR), AI driven drug discovery.

ML in financial services: Key financial applications in machine learning, Fraud detection using anomaly detection, credit risk assessment, Predictive modelling for stock price forecasting, AI driven personalized banking.

ML for Industrial Cybersecurity: Industrial cybersecurity – Threats, Challenges and attacks, supervised learning for malware classification, unsupervised learning for anomaly detection, role of AI in threat intelligence and automated response systems.

**Unit 3. ML for Smart Manufacturing, Retail and Business Intelligence**

**15 Lecture Hours**

Smart Manufacturing with ML: Role of AI and ML in manufacturing industry, Handling different types of data in manufacturing industry, Digital twins and AI driven simulations, ML applications for visual inspection and defect detection, ML for equipment health monitoring.

ML in retail and customer analytics: Key technologies in retail industry, Importance of data driven decision making in retail, Understanding customer segmentation using ML (Clustering Techniques), churn prediction and retention strategies with ML, ML for dynamic pricing and real time price adjustments, Predictive analytics for sales and demand forecasting.

ML for Business Intelligence: Overview of BI and DSS, AI powered dash boards and BI tools, Real time data visualization, performance monitoring, report generation with AI powered insights, Predicting future business trends

**Unit 4. Machines Learning for Predictive Maintenance and Quality Control**

**10 Lecture Hours**

Types of maintenance – Reactive, Preventive, Predictive and Perfective. Benefits of ML based predictive maintenance, techniques for predictive maintenance – Time series forecasting models, and anomaly detection.

Quality control using ML: Traditional vs AI-based Quality control techniques, ethical considerations and data security in quality monitoring, ML for risk assessment, role of computer vision in quality control, Image processing techniques for defect identification, deep learning for automated visual inspection, KPI monitoring and AI powered quality scorecards.

**Unit 5. Machine Learning Model Deployment and Future Trends**

**08 Lecture Hours**

ML model deployment in on cloud services, ML model monitoring and maintenance, Need of explainable AI in critical industries, Explainable AI and model interpretability.

Emerging trends in ML for Industries: Federated learning and privacy preserving, Shift from predictive to prescriptive AI in business, AI bias and fairness challenges.

**Total lecture Hours 60**

**References\***

|  |  |
| --- | --- |
| **Textbooks** | 1. Application of machine learning in industries (IBM ICE Publications) |
| **Reference books** | 1. Ramkumar, M., Anil Kumar, T. C. H., Borpatra Gohain, A., & Poonia, A. (2022). Application of Machine Learning in Industries. AG Publishing House. ISBN: 978-9395468558. 2. Ramasubramanian, K., & Singh, A. (2019). Machine Learning Using R: With Time Series and Industry-Based Use Cases in R (2nd ed.). Apress. ISBN: 978-1484242148. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Application of Machine Learning in Industries Lab**

**List of Lab Experiments**

**Experiment 1:**

Given a raw industrial dataset with missing values, categorical variables, and outliers, perform data preprocessing, including missing value imputation, encoding categorical data, and feature scaling.

**Experiment 2:**

Implement an anomaly detection algorithm to identify unusual patterns in industrial data of your choice and compare different anomaly detection techniques

**Experiment 3:**

Build a machine learning model to classify financial transactions as "fraudulent" or "legitimate" using a fraud detection dataset and compare at least any 2 techniques.

**Experiment 4:**

**Train a machine learning model to predict whether a patient is at risk of heart disease based on medical data.**

**Experiment 5:**

**Build a deep learning model to perform image classification for the detection of any disease.**

**Experiment 6:**

**Train a deep learning-based image classification model to detect defective products in a manufacturing pipeline. Use a dataset of product images with labeled "defective" and "non-defective" samples.**

**Experiment 7:**

**Develop a time-series forecasting model to predict future sales demand using historical retail sales data. Compare the performance of ARIMA and XGBoost based regression models.**

**Experiment 8:**

**Train a regression model to predict industrial energy consumption based on historical data and apply feature selection techniques to improve the performance.**

**Experiment 9:**

**Build an ensemble model for the identification of a particular disease. Use SHAP values to explain model predictions.**

**Experiment 10:**

**Develop a machine learning model to predict crop yield based on multiple factors like soil quality, weather conditions etc and compare the performance of Decision Trees and Random Forest.**

**Experiment 11:**

**Perform K-Means clustering on a dataset related to industrial processes. Visualize the clusters and evaluate their quality using silhouette scores.**

**Experiment 12:**

**Train a Random Forest and Gradient Boosting model for a classification problem.**

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Application of machine learning in industries (IBM ICE Publications) |
| **Reference books** | 1. Aurélien Géron (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly; Third edition. ISBN: 9355421982 2. Jamie Chan. (2024). Machine Learning with Python for Beginners: A Step-By-Step Guide with Hands-On Projects, Shroff Publishers; First Edition. ISBN-10 ‏ : ‎ 9355425740 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Information Theory and coding** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | Basic knowledge of Computer Hardware | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To acquire knowledge about concept of mutual information and entropy in information theory
2. To acquire knowledge about various data compression codes
3. To understand and analyse the various error correcting codes used for reliable transfer of data.
4. To understand and analyse the decoding techniques

**Course Outcomes**

After completion of course, students would be able to:

1. To apply the several source coding techniques.
2. To implement the several Noisy coding techniques.
3. To implement the channel coding theorem & various codes.
4. To apply Block control coding and error coding techniques.
5. To implement decoding of various codes.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **3** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **3** | **-** |
| **CO 2** | **3** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **3** | **-** |
| **CO 3** | **1** | **2** | **-** | **-** | **-** | **-** | **-** | **1** | **-** | **-** | **2** | **2** | **1** |
| **CO 4** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **-** | **-** | **3** | **3** | **1** |
| **CO 5** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **-** | **-** | **3** | **3** | **1** |
| **Average** | **1.25** | **1.3** |  |  |  |  |  | **1.3** |  |  | **2.6** | **2.8** | **0.6** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: SOURCE CODING 12 Lecture Hours**

Mathematical model for information source-Mutual Information-Discrete Entropy-Definition and properties-Joint and conditional entropies-Entropy in the continuous case-Unique decipherability and Instantaneous codes-Kraft inequality,

**Unit II: NOISY CODING 12 Lecture Hours**

Discrete memory less channel-Classification of channels & channel capacity-Calculation of channel capacity, Decoding schemes-Fano’s Inequality-Shannon’s fundamental theorem-Capacity of a band limited Gaussian channel, Elias code, Arithmetic coding and universal coding.

**Unit III: CHANNEL CODING 10 Lecture Hours**

Channel models: Binary Symmetric channels-Information capacity theorem-Implication of the information capacity theorem-Information capacity of coloured noise channel-Rate distortion theory-Data compression

**Unit IV: ERROR CODING 14 Lecture Hours**

Linear block codes:-Cyclic codes, BCH codes, RS codes, Golay codes, Burst error correcting codes, Interleaved codes, Convolution codes. Convolutional encoder, code tree, state diagram, trellis diagram-Turbo codes.

**Unit V: DECODING OF CODES 12 Lecture Hours**

Maximum likelihood decoding of convolution codes-Sequential decoding of convolution codes-Applications of Viterbi decoding, Exact techniques of decoding, relationship between complexity of algorithms in poly-digital circuits and VLSI with algebraic coding.

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Simon Haykin,”Communication Systems”, John Willey & Sons, Inc, Newyork, 4th Edition.  2. Blahut, R.E, Theory and practice of error control codes, Addison Wesley, 1st Edition. |
| **Reference books** | 1. John G.Proakias,”Digital Communication”, McGraw Hill, Singapore, 4 th Edition, 2001.  2. Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw-Hill publication, 2nd Edition, 2010.  3. James V Stone, Information Theory: A Tutorial introduction, Sebtel Press, 1st Edition, 2015.  4. Thomas M Cover and Joy A Thomas, Elements of Information Theory, Wiley India, 2nd Edition, 2006.   1. Hwei P Hsu,”Theory of Analog and Digital Communication, “Pearson/Prentice Hall, New Jersey   6. Shu Lin & Daniel J.Costello, “Error control coding Fundamentals and applications,” Pearson Education 2nd edition.  7. S.P.Eugene Xavier, “Statistical Theory of Communication,” New Age International, 1997. |
| **Web Resources** | 1. http://www.tekelec.com/ss7/protocols/atm15.asp 2. <http://www.atmforum.com> 3. [www.scribd.com](http://www.scribd.com) 4. www.wikipedia.org |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**List of Experiments Total Lab hours 30**

1. **LAB EXERCISE 1**

Implement various entropies and mutual information of a given channel. Test various types of channel such as:

a) Noise free channel  
b) Error free channel  
c) Binary symmetric channel  
d) Noisy channel  
e) Compare channel capacity of above channels.

**2. LAB EXERCISE 2**

Implement a program for generation and evaluation of variable length source coding using Huffman Coding and decoding (C/MATLAB).

**3. LAB EXERCISE 3**

Implement coding and decoding of Cyclic codes (C/MATLAB).

**4. LAB EXERCISE 4**

Implement coding and decoding of Linear block codes (C/MATLAB).

**5. LAB EXERCISE 5**

Implement coding and decoding of BCH and RS codes (C/MATLAB).

**6. LAB EXERCISE 6**

Implement coding and decoding of Convolutional codes (C/MATLAB).

**7. LAB EXERCISE 7**

Write a simulation program to implement source coding and channel coding for transmitting a text file.

**8. LAB EXERCISE 8**

Implement a program to study performance of a coded and uncoded communication system (calculate the error probability).

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Simon Haykin,”Communication Systems”, John Willey & Sons, Inc, Newyork, 4th Edition.  2. Blahut, R.E, Theory and practice of error control codes, Addison Wesley, 1st Edition. |
| **Reference books** | 1. John G.Proakias,”Digital Communication”, McGraw Hill, Singapore, 4 th Edition, 2001.  2. Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw-Hill publication, 2nd Edition, 2010.  3. James V Stone, Information Theory: A Tutorial introduction, Sebtel Press, 1st Edition, 2015.  4. Thomas M Cover and Joy A Thomas, Elements of Information Theory, Wiley India, 2nd Edition, 2006.   1. Hwei P Hsu,”Theory of Analog and Digital Communication, “Pearson/Prentice Hall, New Jersey   6. Shu Lin & Daniel J.Costello, “Error control coding Fundamentals and applications,” Pearson Education 2nd edition.  7. S.P.Eugene Xavier, “Statistical Theory of Communication,” New Age International, 1997. |
| **Web Resources** | 1. http://www.tekelec.com/ss7/protocols/atm15.asp 2. <http://www.atmforum.com> 3. [www.scribd.com](http://www.scribd.com) 4. www.wikipedia.org |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Signal Processing** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Linear Algebra, Digital Electronics** | | **Syllabus version: 1.0** | | | |

#### **Course Objectives**

1. Introduce fundamental DSP concepts, including frequency representation, discrete-time signals, and LTI systems.
2. Develop an understanding of Z-transform, DFT, and FFT for efficient signal processing.
3. Explore the design and analysis of IIR and FIR digital filters.
4. Study quantization effects, rounding errors, and multi-rate DSP concepts like decimation and interpolation.
5. Apply DSP techniques in speech, audio, image processing, and machine learning applications.

#### **Course Outcomes**

After completion of the course, students will be able to:  
**CO1:** Analyze discrete-time signals, sampling theorem, and LTI systems using Z-

Transforms.  
**CO2:** Apply DFT and FFT for spectral analysis and signal processing.  
**CO3:** Design and implement IIR and FIR filters using various techniques.  
**CO4:** Evaluate quantization effects, round-off errors, and multi-rate DSP methods.  
**CO5:** Utilize DSP techniques in speech, image processing, real-time systems, and ML.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **1** | 1 |
| **CO2** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 1 |
| **CO3** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 2 |
| **CO4** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 2 |
| **CO4** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **Average** | 2 | 1.4 | 1.2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.6 | 1.6 | 1.8 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**UNIT- I: Basic elements of digital signal Processing & Introduction to DFT**

**15 Lecture Hours**

Concept of frequency in continuous time and discrete time signals –Sampling theorem – Discrete time signals. Discrete time systems –Analysis of Linear time invariant systems –Z transform –Convolution and correlation, Efficient computation of DFT Properties of DFT – FFT algorithms – Radix-2 and Radix-4 FFT algorithms – Decimation in Time – Decimation in Frequency algorithms – Use of FFT algorithms in Linear Filtering and correlation.

**UNIT- II Structure of IIR 10 Lecture Hours**

System Design of Discrete time IIR filter from continuous time filter – IIR filter design by Impulse Invariance. Bilinear transformation – Approximation derivatives – Design of IIR filter in the Frequency domain.

**UNIT – III Symmetric & Anti-symmetric FIR filters 15 Lecture Hours**

Linear phase filter – Windowing techniques – rectangular, triangular, Blackman and Kaiser windows – Frequency sampling techniques – Structure for FIR systems.

**UNIT-IV Finite word length effects in FIR and IIR digital filters 15 Lecture Hours**

Quantization, round off errors and overflow errors. Multi rate digital signal processing: Concepts, design of practical sampling rate converters, Decimators, interpolators. Polyphasedecompositions.Application of DSP – Model of Speech Wave Form – Vocoder

### **UNIT-V Applications of DSP in Computer Science 5 Lecture Hours**

### **Speech and Audio Processing, Image Processing using DSP, Machine Learning & DSP, Real-time DSP Systems**

**Total lecture Hours 60**

# References

|  |  |
| --- | --- |
| **Textbooks** | 1. Oppenheim and R. W. Shafer, Discrete-Time Signal Processing, 3rd edition. Pearson, 2014. 2. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4th edition. Pearson, 2007. |
| **Reference books** | 1. K. Mitra, Digital Signal Processing: A computer-Based Approach, 3rd edition. Tata McGraw-Hill, 2006. 2. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice Hall India, 2005. 3. Antoniou, Digital Filters: Analysis, Design and Applications. Tata McGraw-Hill, New Delhi, 2003. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Signal Processing Lab**

**List of Experiments**

### **Experiment No 1** Generate and visualize discrete-time signals.

### **Experiment No 2** Understand convolution and correlation properties.

### **Experiment No 3** Analyze signals in the frequency domain using DFT and FFT.

### **Experiment No 4** Analyze discrete-time systems using Z-transform.

### **Experiment No 5** Design and implement IIR filters.

### **Experiment No 6** Implement FIR filters using windowing techniques.

### **Experiment No 7** Understand quantization, rounding errors in digital filters and **Finite**

### **Word Length Effects in DSP**

### **Experiment No 8** Implement decimation and interpolation techniques.

### **Experiment No 9** Analyze speech signals using DSP techniques.

**Experiment No 10** Apply DSP concepts to image processing.

**Experiment No 11** Use DSP techniques for feature extraction in ML applications.

### **Software & Tools for Implementation:**

**Python (NumPy, SciPy, Matplotlib, librosa, OpenCV for image processing)**

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Learning OpenCV 3: Computer Vision with OpenCV Library" – Gary Bradski & Adrian Kaehler 2. Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Scene Classification" – Himanshu Singh |
| **Reference books** | 1. Discrete-Time Signal Processing by Alan V. Oppenheim and Ronald W. Schafer 2. Digital Signal Processing: Principles, Algorithms, and Applications by John G. Proakis and Dimitris K. Manolakis 3. Digital Signal Processing: A Computer-Based Approach by Sanjit K. Mitra 4. Understanding Digital Signal Processing by Richard G. Lyons 5. The Scientist and Engineer's Guide to Digital Signal Processing by Steven W. Smith |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Real Time operating systems** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Microprocessor and Embedded Systems,Operating Systems** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1.To understand the basics of operating systems tasks and basic OS architectures

2. To understand concepts of task scheduling

3.To analyse the design issues pertaining to multitasking

4.To learn interfacing memory and I/O with RTOS kernels

5.To develop the knowledge required for designing software related to embedded systems using a real-time operating system.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1** understand and develop basic OS architectures and RTOS

**CO2** Apply the concepts of task scheduling and multitasking

**CO3** Interface memory and I/O with RTOS kernel.

**CO4** Develop strategies for Memory management in RTOS

**CO5** Develop embedded systems using RTOS

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | 1 |
| **CO 2** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 3 | 2 |
| **CO 3** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 | 1 |
| **CO 4** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 | 1 |
| **CO 5** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 | 1 |
| **Average** | 2 | 1.4 | 1.2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2.6 | 3 | 1.2 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**:UNIT I**

**Fundamentals of operating systems: 8 Lecture Hours**

Operating system objectives and functions, Virtual Computers,Interaction of O. S. & hardware architecture, Evolution of operating systems ,Architectures of OS , Multi programming, Multitasking, Multiuser, parallel,distributed & RTOS.

**UNIT-II**

**Scheduling: 12** **Lecture Hours**

Uniprocessor Scheduling: Types of scheduling Scheduling algorithms: FCFS, SJF, Priority, Round Robin queue scheduling, Thread Scheduling, Multiprocessor Scheduling concept.Concurrency: Principles of Concurrency, Mutual Exclusion H/W Support, software approaches, Semaphores and Mutex,Message Passing techniques

**UNIT III**

**Synchronization : 15** **Lecture Hours**

Synchronization: Readers-Writers Problem, Producer Consumer Problem, Dining Philosopher problem.Deadlock: Principles of deadlock, Deadlock Prevention,Deadlock Avoidance, Deadlock Detection, An Integrated Deadlock Strategies.

**UNIT IV 15 Lecture Hours**

**Memory Management:**

Memory Management requirements, Memory partitioning:Fixed, dynamic, partitioning Memory allocation Strategies , Fragmentation, Swapping, Segmentation, Paging,Virtual Memory, Demand paging Page Replacement Policies ,Thrashing, Working Set Model

**UNIT V 10 Lecture Hours**

**I/O Management:**

I/O Management and Disk Scheduling: I/O Devices,Organization of I/O functions Operating System Design issues, I/O Buffering, Disk Scheduling , study of RTOS: Vxworks and μCOS

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Sam Siewert, V, Real-Time Embedded Components and Systems: With Linux and RTOS(Engineering), 2015  2.C.M. Krishna and G.Shin, Real Time Systems, McGraw-Hill International Edition, |
| **Reference books** | 1. Tanenbaum, Modern Operating Systems, 3/e, Pearson Edition, 2007. 2. 2. VxWorks: Programmer's Guide 5.4, Windriver, 1999 3. Wayne Wolf, Computers as Components: Principles of Embedded Computing System   Design, 2/e, Kindle Publishers, 2005. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** | NPTEL |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Real Time operating systems Labs**

**List of Experiments**

**Overview of LINUX as Real Time operating Systems**

1.Write the pseudo code in Linux using C/C++ to perform FCFS scheduling

**2** Write the pseudo code in Linux using C/C++ to perform Round Robin scheduling

**3** Write the pseudo code in Linux using C/C++ to perform Priority Based scheduling

**4** Write the pseudo code in Linux using C/C++ to perform Print parent process ID & child process ID using Fork()

**5** Study the POSIX thread & Write the pseudo code in Linux using C/C++

**6** Study of Semaphore & Write appropriate the pseudo code in Linux using C/C++

**7** Write appropriate pseudo code for blinking of LED and keypad in Linux using python

**8** Write pseudo code for pipe in Linux

**9** Study of Dining Table philosophy problem and write appropriate pseudo code for the same

1. Write an application that creates two tasks of the same priority and sets the time slice period to illustrate time slicing.
2. Write an application that creates a two task to Blinking two different LEDs at different timings
3. Write an application that creates a two task to Blinking two different LEDs at different timings
4. Write an application that creates a two task displaying two different messages in LCD display in two lines.
5. Write an application for Sending messages to mailbox by one task and reading the message from mailbox by another task. 10 Sending message

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | Sam Siewert, V, Real-Time Embedded Components and Systems: With Linux and RTOS(Engineering), 2015 |
| **Reference books** | C.M. Krishna and G.Shin, Real Time Systems, McGraw-Hill International Edition |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Introduction to Cloud Computing Architecture and Design** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To make students able to understand the impact of cloud based applications on business.
2. To learn students the way to collect relevant data from different sources.
3. To enable students understand various algorithms for processing data in effective manner using cloud platform and services

**Course Outcomes**

After completion of course, students would be able to:

**CO1:** Analyze and compare the key characteristics, benefits, and limitations of IaaS, PaaS, and SaaS cloud service models to effectively assess their suitability for different organizational needs.

**CO2:** Analyze and assess the cloud deployment models (Public, Private, Hybrid) demonstrating a deep understanding of their advantages, limitations, and challenges to make informed decisions regarding the selection and management.

**CO3:** Analyze and evaluate NIST, IBM's and AWS cloud computing reference architectures to make informed decisions about their adoption and implementation in diverse usage scenarios.

**CO4:** Ability to apply fundamental as well as advanced cloud computing architectures in various business scenarios.

**CO5:** Design, deploy, and manage scalable and secure cloud-based solutions by leveraging key cloud computing principles.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | - | 1 | - |
| **CO 2** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | - | - | 1 |
| **CO 3** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | - | - | 1 |
| **CO4** | 1 | 1 | **-** | 1 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | 1 | - | - | - |
| **CO5** | 1 | - | 1 | - | - | **-** | **-** | **-** | **-** | **-** | **-** | 1 |  | 1 | - |
| **Average** | 1 | 0.8 | **0.2** | 0.8 | **1.6** | **-** | **-** | **-** | **-** | **-** | **-** | **1** | - | 0.4 | 0.4 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**UNIT 1**  **10 lecture hours**

**Introduction to Virtualization**

Traditional IT Infrastructure, Benefits of Virtualization, Types of Virtualization, History of Virtualization, Types of Server Virtualization, Hypervisors, Anatomy of Server Virtualization, Benefits of Storage Virtualization, Types of Storage Virtualization, VPN, VLAN,  Benefits of Application Virtualization.

**UNIT 2 10 lecture hours**

**Introduction to** Cloud Computing

History, Definition and Characteristics of Cloud computing, Importance of Virtualization in Cloud, Anatomy of Cloud, Cloud deployment models (Public, Private, Community and Hybrid), Cloud delivery models (IaaS, PaaS and Saas), stepping stones for the development of cloud, Grid Computing vs Cloud Computing.

**Unit 3. Cloud Computing Reference Architectures (NIST & IBM) 15 lecture hours**

***NIST Cloud Computing Reference Architecture (CCRA):*** Objectives of NIST, The conceptual reference model, Example: Usage scenarios, Cloud consumer, Cloud provider, Cloud auditor, Cloud broker, Cloud carrier, Scope of control between provider and consumer, CCRA: Architectural components, Service orchestration, Cloud service management, Business support, Provisioning and configuration Portability and interoperability, Security, Privacy, Cloud taxonomy.

***IBM’s CCRA:*** IBM CCRA roles, Cloud service consumer, Cloud service provider, Cloud services, Infrastructure, Common Cloud Management Platform (CCMP), CCMP supports any level of virtualization, Business Support Services (BSS), Operational Support Services (OSS), Security, resilience, performance and consumption, Cloud service creator: Service development tools, IBM CCRA versions or CCRA evolution, Adoption patterns, Adoption pattern in CCRA 3.0, Examples of cloud services.

**Unit 4. Fundamental Cloud Architectures 10 lecture hours**

*Fundamental Cloud Architectures*: Workload distribution architecture, Resource pooling architecture, Dynamic scalability architecture, Elastic resource capacity architecture, Service load balancing architecture, Cloud bursting architecture, Elastic disk provisioning architecture, Redundant storage architecture.

**Unit 5. Advanced Cloud architecture and AWS CCRA 15 lecture hours**

***Advanced Cloud Architectures*:** Overview of the advanced cloud architecture, Hypervisor clustering architecture, Load balanced virtual server instances architecture, Non-disruptive service relocation architecture, Zero downtime architecture, Cloud balancing architecture, Resource reservation architecture, Dynamic failure detection and recovery, Bare-metal provisioning architecture, Rapid provisioning architecture, Components that can comprise the system, Automated administration pattern, Storage workload management architecture, Live VM migration.

***AWS CCRA:*** What is amazon web services, Features of AWS, Web application hosting, Content and media serving, Large scale computing and huge data sets, Disaster recovery for local applications, Ecommerce website: Web frontend, Ecommerce website: Checkout service, Marketing and recommendations, Fault tolerance and high availability, File synchronization service

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Cloud Computing Architecture (IBM ICE Publication) 2. Cloud Computing Deployment Model (IBM ICE Publication) 3. Cloud Computing For Dummies Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper, For Dummies, Edition 1, Nov 2009. 4. Cloud Computing: Concepts, Technology & Architecture, Thomas Erl,Pearson Education India, Jan 2014 5. OpenStack Essentials, Dan Radez, PackIT publication, publications , 2nd Edition, Jan 2016 |
| **Reference books** | 1. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley, 2013. 2. Gautam Shroff, “Enterprise Cloud Computing Technology Architecture Applications”, Cambridge, 2010. 3. , “About Openstack: A Comprehensive Tutorial To Revolutionize Cloud Computing Solutions”, Independently published, March 27, 2023. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**CCA and Design Lab**

**List of Experiments** Study of major cloud computing platforms such as AWS, GCP and Azure.

1. Introduction to AWS Identity and Access Management (IAM)
2. S3: Multi-region Storage Backup with Cross-Region Replication
3. Introduction to Amazon DynamoDB
4. Introduction to EC2 instance and EBS.
5. Introduction to Amazon API Gateway
6. Introduction to Amazon CloudFront
7. Introduction to Amazon Virtual Private Cloud (VPC)
8. Introduction to Amazon EC2 Auto Scaling
9. Introduction to AWS Lambda
10. Installing LAMP Stack on an EC2 Instance

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1.IBM Material Cloud Deployment Model Lab 2. Theo H King, “The Ultimate Guide From Beginners To Advanced For The Amazon Web Services”, Independently Published, 2020. 3. Kevin Jackson, Cody Bunch, and Egle Sigler, “OpenStack Cloud Computing Cookbook”, 3rd Edition, Packt, 2015. |
| **Reference books** | 1. “AWS Cookbook”, 1st Edition, O'Reilly Media, 2021. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Graph Theory and Applications** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Data Structures and Algorithms** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To introduce the fundamental concepts and terminologies of graph theory.
2. To develop the ability to analyze and implement graph traversal and search algorithms.
3. To understand and apply graph-based algorithms for shortest paths, spanning trees, and network flow.
4. To explore graph coloring, matching, and covering techniques for real-world applications.
5. To study advanced topics such as planar graphs, random graphs, and their applications in computing.

**Course Outcomes**

After completion of course, students would be able to:

**CO 1**: Demonstrate a strong understanding of fundamental graph theory concepts and structures.

**CO 2:** Implement and analyze graph traversal algorithms such as BFS and DFS for various applications.

**CO 3:** Apply shortest path and spanning tree algorithms in networking, routing, and optimization problems.

**CO 4**: Solve real-world problems using graph coloring, matching, and covering techniques.

**CO 5:** Explore advanced graph theory applications in areas such as artificial intelligence, blockchain, and data science.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Outcomes** | PO  1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO  9 | PO  10 | PO  11 | PO  12 | PSO  1 | PSO  2 | PSO  3 |
| CO1 | 1 | - | 2 | - | 1 | - | - | - | - | - | - | 1 | 2 | - | - |
| CO2 | 2 | 1 | 1 | - | 1 | - | - | - | - | - | - | 1 | - | - | - |
| CO3 | 2 | 1 | 2 | 1 | 1 | - | - | - | - | - | - | 1 | 1 | - | - |
| CO4 | 2 | 1 | 2 | 1 | 1 | - | - | - | - | - | - | 1 | 1 | - | 1 |
| CO5 | 2 | 2 | 3 | 2 | 2 | 2 | - | - | - | - | - | 2 | - | - | 2 |
| Average | 1.8 | 1 | 2 | 0.8 | 1.2 | 0.2 | - | - | - | - | - | 1.2 | 0.8 | - | 0.6 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

#### **Unit 1: Introduction to Graph Theory 15 lecture hours**

Fundamentals of Graphs: **Definitions, Types (Directed, Undirected, Weighted, Unweighted),** Graph Representation: **Adjacency Matrix, Adjacency List, Incidence Matrix,** Special Graphs: **Complete Graphs, Bipartite Graphs, Trees, Cycles, Paths,** Graph Isomorphism & Subgraphs,

#### **Unit 2: Graph Traversal and Connectivity 15 lecture hours**

**Graph Traversal Algorithms**: Breadth-First Search (BFS), Depth-First Search (DFS), Connected & Disconnected Graphs, **Strongly Connected Components (SCC)** and **Bi-connected Components, Eulerian and Hamiltonian Graphs,** Applications: **Web Crawling, Social Network Analysis**

#### **Unit 3: Trees and Spanning Trees 10 lecture hours**

Properties of **Trees & Binary Trees, Minimum Spanning Tree (MST)**: Prim’s Algorithm, Kruskal’s Algorithm, **Shortest Path Algorithms**: Dijkstra’s Algorithm, Bellman-Ford Algorithm, **Applications in Computer Networks & Distributed Systems**

#### **Unit 4: Graph Coloring, Matching, and Covering 10 lecture hours**

**Graph Coloring**: Chromatic Number, Applications in Scheduling, **Matching Theory**: Maximum Matching, Hall’s Marriage Theorem, **Vertex Cover, Edge Cover, Independent Sets,** Applications: **Register Allocation, Map Coloring, Job Scheduling**

#### **Unit 5: Advanced Topics & Applications 10 lecture hours**

**Planar Graphs**: Euler’s Formula, Kuratowski’s Theorem, **Network Flow Algorithms**: Max Flow – Min Cut Theorem, Ford-Fulkerson Algorithm, **Random Graphs: Erdős–Rényi Model, Small-World Networks, Scale-Free Networks, Applications of random graphs in social networks, Probabilistic Graph Theory, Applications in AI, Machine Learning, and Blockchain**

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Edgar G. Goodaire and Michael M. Parmenter, Discrete Mathematics with Graph Theory, 2nd Edition, Pearson Education (Singapore) P. Ltd., Indian Reprint 2003. |
| **Reference books** | 1. Rosen, Kenneth H. *Discrete mathematics & applications*. McGraw-Hill, 1999. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Graph Theory Lab**

**List of Experiments**

1. **Graph Representation & Visualization**
   * Implement Adjacency Matrix and Adjacency List representations of graphs.
   * Visualize graphs using Matplotlib (Python) or NetworkX.
2. **Graph Traversal Algorithms**
   * Implement Breadth-First Search (BFS) and Depth-First Search (DFS).
   * Identify connected components in a graph.
3. **Shortest Path Algorithms**
   * Implement Dijkstra’s Algorithm for single-source shortest path.
   * Implement Bellman-Ford Algorithm for graphs with negative weights.
4. **Minimum Spanning Tree (MST)**
   * Implement Prim’s Algorithm for MST.
   * Implement Kruskal’s Algorithm using Disjoint Set Union (DSU).
5. **Eulerian and Hamiltonian Graphs**
   * Check if a given graph is Eulerian or Hamiltonian.
   * Implement Fleury’s Algorithm for finding Eulerian circuits.
6. **Graph Coloring**
   * Implement Greedy Coloring Algorithm to determine chromatic number.
   * Solve map coloring problems using graph coloring techniques.
7. **Network Flow Algorithms**
   * Implement Ford-Fulkerson Algorithm for Maximum Flow.
   * Apply Max Flow – Min Cut Theorem in a given network.
8. **Matching and Covering Problems**
   * Implement Bipartite Graph Check using BFS.
   * Implement Maximum Bipartite Matching using Hungarian Algorithm.
9. **Planar Graphs and Euler’s Formula**
   * Verify Euler’s Formula (V - E + F = 2) for given graphs.
   * Check for graph planarity using Kuratowski’s Theorem.
10. **Graph Applications in Real-World Scenarios**

* Implement PageRank Algorithm for web ranking.
* Apply Social Network Analysis (SNA) techniques on real-world datasets.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Edgar G. Goodaire and Michael M. Parmenter, Discrete Mathematics with Graph Theory, 2nd Edition, Pearson Education (Singapore) P. Ltd., Indian Reprint 2003. |
| **Reference books** | 1. Rosen, Kenneth H. *Discrete mathematics & applications*. McGraw-Hill, 1999. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

### **Course Objectives**

1. To introduce fundamental concepts of digital image processing, including image representation, enhancement, and transformation techniques.
2. To develop an understanding of image restoration, noise characterization, and various restoration techniques.
3. To explore image compression methods, including lossy and lossless techniques, entropy coding, and wavelet-based compression.
4. To study morphological image processing techniques for feature extraction and image segmentation methods for object detection.
5. To apply image processing concepts to real-world applications, including image filtering, edge detection, and digital watermarking.

### **Course Outcomes**

After completion of this course, students will be able to:  
**CO1:** Analyze and manipulate digital images using various enhancement and

transformation techniques.  
**CO2:** Implement image restoration methods to recover degraded images and apply

noise filtering techniques.  
**CO3:** Apply image compression techniques, including Huffman coding, DCT, and

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Digital Image Processing** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Linear Algebra, DSA, Statistics, Probability** | | **Syllabus version: 1.0** | | | |

wavelet-based compression.  
**CO4:** Utilize morphological image processing techniques for object detection and

feature extraction.  
**CO5:** Implement image segmentation algorithms such as thresholding, edge detection,

and region-based segmentation for various applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **1** | 1 |
| **CO2** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 1 |
| **CO3** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **CO4** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 2 | 3 |
| **Average** | 2 | 1.4 | 1.2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.4 | 1.6 | 2.2 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Unit I: Image representation and analysis 12 Lecture Hours**

Introduction to Digital Image Processing, Numerical representation of images, Image augmentation, enhancement, processing, color transforms, geometric transforms, feature recognition and extraction, Eigen values and eigen vectors, Rank of matrix and SVD, Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization, Contrast stretching, Histogram equalization, Correlation and convolution, Smoothing filters, Sharpening filters, gradient and Laplacian, Frequency domain filtering,Hotelling Transform, Fourier Transforms and properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform.

**Unit II: Image Restoration 8 Lecture Hours**

Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections.

**Unit III: Image Compression 20 Lecture Hours**

Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding, FAX compression (CCITT Group-3 and Group-4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation Wavelet based Image Compression, Wavelet based Image Compression, Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG2000 encoding, Digital Image Watermarking.

**Unit IV: Morphological Image Processing 10 Lecture Hours**

Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion.

**Unit V: Image Segmentation 10 Lecture Hours**

Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's method, Moving averages, Multivariable thresholding, Regionbased segmentation, Watershed algorithm, Use of motion in segmentation

**Total lecture Hours 60**

# References

|  |  |
| --- | --- |
| **Textbooks** | 1. Digital Image processing by Rafael c Gonzalez & Richard E Woods, 3rd Edition 2. Digital Image Processing by William K Pratt |
| **Reference books** | 1. Richard O. Duda, Peter E. Hard, David G. Stork, Pattern Recognition, 2nd, Wiley, 2021. 2. Prince SJ. Computer vision: models, learning, and inference. Cambridge University Press; 2012 Jun 18. 3. Theodoridis S, Koutroumbas K. Pattern recognition. Elsevier; 2006 Apr 7. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Digital Image Processing Lab**

**List of Experiments**

**Experiment No 1** Simulation and Display of an Image, Negative of an Image(Binary &

Gray Scale)

**Experiment No 2** Implementation of Relationships between Pixels

**Experiment No 3** Implementation of Transformations of an Image

**Experiment No 4** Contrast stretching of a low contrast image, Histogram, and

Histogram Equalization

**Experiment No 5** Display of bit planes of an Image

**Experiment No 6** Display of FFT(1-D & 2-D) of an image

**Experiment No 7** Computation of Mean, Standard Deviation, Correlation coefficient of

the given Image

**Experiment No 8** Implementation of Image Smoothening Filters(Mean and Median

filtering of an Image)

**Experiment No 9** Implementation of image sharpening filters and Edge Detection using

Gradient Filters

**Experiment No 10** Image Compression by DCT,DPCM, HUFFMAN coding

**Experiment No 11** Implementation of image restoring techniques

**Experiment No 12** Implementation of Image Intensity slicing technique for image

enhancement

**Experiment No 13** Canny edge detection Algorithm

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Learning OpenCV 3: Computer Vision with OpenCV Library" – Gary Bradski & Adrian Kaehler 2. Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Scene Classification" – Himanshu Singh |
| **Reference books** | 1. Digital Image Processing: An Algorithmic Approach" – Wilhelm Burger & Mark J. Burge 2. Computer Vision: Algorithms and Applications" – Richard Szeliski 3. Handbook of Image and Video Processing" – Al Bovik |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Microcontroller and Embedded Systems** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Digital Electronics, Introduction to C** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Analyse the functional blocks of 8085 and 8086 microprocessors
2. Design the hardware using different styles of modeling for targeting 8051 Microcontroller chip and kit
3. Develop the Assembly and Embedded ‘C’ Code for the development of the system
4. Develop embedded systems for system design applications
5. Hardware interfacing and real-life applications of embedded systems.

**Course Outcomes**

On completion of this course, the students will be able to

CO1. Understand the functional modules of general purpose 8bit and 16 bit processors

CO2 Analyse various aspects of ALU and CPU building blocks

CO3. Design the embedded system using microcontrollers such as 8051 MCS, and develop the code using assembly language programming.

CO4. Develop the embedded ‘C’ code for different applications and interfacing units of the microcontroller.

CO5 Analyse and develop for different case studies, specifications and sampled control embedded applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **1** | 1 |
| **CO 2** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 2 |
| **CO 3** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 | 2 |
| **CO 4** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 | 2 |
| **CO 5** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 3 | 3 | 2 |
| **Average** | 2 | 1.4 | 1.2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2.4 | 2.4 | 1.8 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**UNIT I: Microprocessor fundamentals (8bits) 10 Lecture Hours**

# Introduction to Microprocessor – Architecture of Microprocessor 8085-Internal registers (8-bit & 16-bit)-CPU-ALU-Types of System Bus-Bus Structure- multiplexing and demultiplexing address/data bus-Instruction Register and Decoder - Timing and Control Unit-Interrupts and Serial I/O (principle only)-external memory – Block diagram of 8085-Programmer’s model of 8085-pin configuration of 8085.

# UNIT II: Microprocessor fundamentals (16bits) 10 Lecture Hours

8086 PROCESSOR: Historical background , 8086 CPU Architecture ,addressing modes, Machine language instruction formats, Machine coding the program . INSTRUCTION SET OF 8086: Data transfer and arithmetic instructions. Control/Branch Instructions, Illustration of these instructions with example programs .

# UNIT III: Designing ALU and CU 15 Lecture Hours

Machine Instructions, Opcode, Registers, CPU organization, Instruction formats, Timing and control, Instruction cycle, Addressing modes, Program Control, Instruction Cycle: Fetch Decode and Execute, Control Transfer, Control memory, Micro programmed vs. Hardwired control unit

# UNIT IV: MSC 51 Family 15 Lecture Hours

Study of micro controller (MCS-51family- 8051) - Architecture, instruction set, addressing modes and programming, Registers, Flags, Counter and Timers, Comparison of various families of 8-bit micro controllers. Interfacing of ADC, sensors, keyboard and DAC using microcontrollers.

# UNIT V: Embedded System and Program Development Tools 10 Lecture Hours Introduction to Embedded Systems, Embedded System: Categories, Requirements and Design Challenges, embedded computing, Applications Areas, Recent trends in embedded systems, Development process & Design, Formalisms for System Design: Integration and testing, Packaging Configuration, Development tools, Linker, Loader, Compiler, Libraries. Design Tools: Kiel, Arduino. Design Case Examples

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Microprocessor Architecture, Programming and Application with the8085, Ramesh S. Gaonakar, PenramLnternational Publishing, Mumbai, 2. . Microprocessor and Interfacing - Douglas V Hall, SSSP Rao, 3rd edition TMH, . 3. The 8051 Microcontroller, [Kenneth J. Ayala](http://www.google.co.in/search?tbo=p&tbm=bks&q=inauthor%3A%22Kenneth%2BJ.%2BAyala%22) 4. Computers As Components: Principles of Embedded Computing System, [Marilyn Wolf](http://www.google.co.in/search?hl=en&safe=active&sa=G&tbo=d&tbm=bks&tbm=bks&q=inauthor%3A%22Marilyn%2BWolf%22&ei=l20LUarCIcjXrQeQt4C4Aw&ved=0CDwQ9AgwAA) – 2012 |
| **Reference books** | * 1. .Embedded system Architecture programming design, Raj kamal, 2nd edition   2. An Embedded software primer, David E Simon Low price edition. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** | NPTEL |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Microprocessor and Embedded Systems Lab**

**List of Experiments**

1. **Overview of Microprocessors, Microcontrollers and embedded systems**

Addition, Subtraction, multiplication and Division of 8-bit and 16 bit numbers using 8085/8086/8051

* **Install the simulators for the different processors**
* **Write Assembly level programs for arithmatic operations**
* **Execute the Program in simulators and Kit**

1. Sorting of Array of numbers using 8085/8086/8051

* **Install the simulators for the different processors**
* **Write Assembly level programs for sorting algorithms operations**
* **Execute the Program in simulators and Kit**

1. Searching for Character in a String using above processors

* **Install the simulators for the different processors**
* **Write Assembly level programs for searching algorithms operations**
* **Execute the Program in simulators and Kit**

1. String Manipulations Programs for above processor

* **Install the simulators for the different processors**
* **Write Assembly level programs for different algorithms for string operation**
* **Execute the Program in simulators and Kit**

1. Interfacing LED,LCD with the above processors

* **Install the simulators for the different processors**
* **Interface LED and LCD with different processors**
* **Write Assembly level programs for interfacing LED and LCD**
* **Execute the Program in simulators and Kit**

1. Design Digital Clocks using the processors as above

* **Install the simulators for the different processors**
* **Write Assembly level programs for designing Digital clocks**
* **Execute the Program in simulators and Kit**

1. Interfacing ADC&DAC with Microcon troller 8051

* **Install the simulators for 8051**
* **Interface selected ADC and DAC of 8 bits with 8051**
* **Write Assembly level programs for interfacing ADC and DAC**
* **Execute the Program in simulators and Kit**

1. Parallel Communication between Two Microprocessors using 8255

* **Install the simulators for the different processors**
* **Interface 8255 with different processors**
* **Establish communication 0f 8255 with the processors**
* **Execute the Program in simulators and Kit**

1. Arithmetic, Logical and Bit Manipulation Instructions of 8051

* **Install the simulators for 8051**
* **Write Assembly level programs for bit manipulation using 8051 instruction set**
* **Execute the Program in simulators and Kit**

1. Interfacing Timer/Counters in 8051

* **Install the simulators for 8051**
* **Interface the given Timer/counter with 8051**
* **Write Assembly level programs for interfacing Timer/counter**
* **Execute the Program in simulators and Kit**

1. . Write a program to set up interrupt process in 8051

* **Install the simulators for the different processors**
* **Write instruction for enabling given inetrrupt process**
* **Execute the Program in simulators and Kit**

1. Interfacing through UART in 8051

* **Install the simulators for the different processors**
* **Interface UART with 8051**
* **Write Assembly level programs for interfacing UART with 8051**
* **Execute the Program in simulators and Kit**

1. Interfacing LCD to 8051

* **Install the simulators for the different processors**
* **Interface LCD with 8051**
* **Write Assembly level programs for interfacing LCD with 8051**
* **Execute the Program in simulators and Kit**

1. Interfacing Matrix keyboard to 8051

* **Install the simulators for the different processors**
* **Interface keyboard of given dimension with different 8051**
* **Write Assembly level programs for interfacing keyboard**
* **Execute the Program in simulators and Kit**

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | Sam Siewert, V, Real-Time Embedded Components and Systems: With Linux and RTOS(Engineering), 2015 |
| **Reference books** | C.M. Krishna and G.Shin, Real Time Systems, McGraw-Hill International Edition |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Big Data Overview** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Database Management Systems, Python, Java Programming** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the fundamental concepts of Big Data, including its characteristics, storage mechanisms, and computing trends, and recognize its significance in modern data-driven applications.
2. To explore various Big Data storage architectures, technologies, and management strategies, including distributed storage systems, cloud-based storage, and security considerations.
3. To analyze large-scale datasets using Big Data processing frameworks such as Hadoop and Apache Spark, and apply machine learning techniques for data-driven insights.
4. To develop proficiency in data visualization techniques and tools for effectively interpreting and presenting large and complex datasets in real-time and interactive formats.
5. To examine real-world applications and industry case studies of Big Data analytics, exploring emerging trends, challenges, and best practices for future advancements.

**Course Outcomes**

On completion of this course, the students will be able to

**CO1**: Understand the concept of Data and Big Data.

**CO2**: Explain various components of Big Data Ecosystem.

**CO3**: Exploring the real-world use cases and applications of Big Data.

**CO4**: Understanding of management principles and techniques, of handling Big Data

**CO5**: Developing practical skills in setting up big data ecosystems, working with

distributed storage, managing data transfers, and handling real-time streaming

applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | **-** | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | **1** | 1 |
| **CO 2** | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 1 |
| **CO 3** | 2 | 2 | 1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 2 |
| **CO 4** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 2 | 2 | 2 |
| **CO 5** | 2 | 2 | 2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1 | 1 | 3 |
| **Average** | 2 | 1.4 | 1.2 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | 1.4 | 1.4 | 1.8 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction to Big Data 10 Lecture Hours**

Data Growth explosion, Categories of Data, Different Data Storage mechanisms, Introduction to a flat file, tabular and relational databases, Introduction to NoSQL data stores, Characteristics of Big data, (Volume, Velocity, Variety, Value, Veracity), Information mining and benefits of big data; Risks of Big Data, Structure of Big Data, Trends of Computing for Big Data, High-performance Computing (Supercomputers and Clusters), Grid Computing, Cloud Computing, Mobile Computing, Big Data Pipleline

**Unit II: Big Data Storage & Big Data Management 15 Lecture Hours**

Big Data Storage vs. Traditional Storage Systems, Challenges in Big Data Storage, Scalability, Reliability, and Performance in Big Data Storage, **Big Data Storage Architectures, Distributed Storage Systems:** Google File System (GFS), Hadoop Distributed File System (HDFS), Amazon S3 & Object Storage, **File Storage vs. Block Storage vs. Object Storage, RAID, SAN, NAS for Big Data, Cloud Storage for Big Data,** AWS S3, Google Cloud Storage, Azure Blob Storage, **Storage Technologies for Big Data: Relational vs. Non-Relational Storage, NoSQL Databases for Big Data Storage, Data Warehousing vs. Data Lakes, Data Storage & Processing Integration, Big Data Processing with Storage:** Apache Hadoop & HDFS, Apache Spark with Storage Backends, **Real-Time Data Storage:** Apache Kafka for Streaming Storage, Time-Series Databases (InfluxDB, TimescaleDB), **Data Storage Optimization Techniques:** Data Partitioning & Indexing, Data Compression Techniques, **Security & Compliance in Big Data Storage.**

**Unit III: Data Analysis**  **15 Lecture Hours**

## **Big Data Processing Frameworks for Analysis, Batch Processing vs. Real-Time Processin, Hadoop Ecosystem for Big Data Analysis, Apache Spark for Data Analysis, NoSQL Databases for Big Data Analysis, Machine Learning Techniques for Big Data Analysis: Supervised Learning, Unsupervised Learning, Time-Series Analysis for Big Dat, Sentiment Analysis & Natural Language Processing (NLP)**

**Unit IV: Data Visualization 15 Lecture Hours**

Importance of Data Visualization in Big Data Analytics, Data Visualization vs. Data Analysis, Data Visualization Libraries, BI & Dashboarding Tools, Cloud-based Visualization Services, High-Dimensional Data Visualization, Geospatial Data Visualization, Time-Series Data Visualization, Network & Graph Visualization, Real-Time & Interactive Visualization, Intoduction to tools and techniques for data visualization.

**Unit V: Real World Applications and Use cases**       **5 Lecture Hours**

Need for and Importance of Big Data Analytics, The Need for Standards; Case studies from various industries (e.g., finance, healthcare, e-commerce), Emerging trends and future directions in big data.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques,   NoSQL, and Graph. By David Loshin, Elsevier,  August 23, 2013.   1. Sqoop: O'Reilly Media |
| **Reference books** | 1. **"Big Data: Principles and Best Practices of Scalable Real-time Data Systems"** – Nathan Marz and James Warren (Manning Publications, 2015) 2. **"Big Data: A Revolution That Will Transform How We Live, Work, and Think"** – Viktor Mayer-Schönberger and Kenneth Cukier (Mariner Books, 2014) 3. **"Sqoop: Data Transfer Between Hadoop and Relational Databases"** – O'Reilly Media (by Jarek Jarcec Cecho, Kathleen Ting, 2015) 4. **"Hadoop in Practice"** – Alex Holmes (Manning Publications, 2014) 5. **"Professional Hadoop Solutions"** – Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich (Wiley, 2013) 6. **"Apache Sqoop Cookbook"** – Kathleen Ting and Jarek Jarcec Cecho (O'Reilly Media, 2013) 7. **"Mastering Hadoop"** – Sandeep Karanth (Packt Publishing, 2017) |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Big Data Overview Lab**

**List of Experiments**

**Experiment 1: Installation of the Cloudera QuickStart virtual machine**

1. Install Cloudera Hadoop Framework on the VM
2. Check the Hadoop tools present and running using HUE

**Experiment 2: Understand Basic Linux and HDFS commands**

1. Demonstrate various Linux commands
2. Demonstrate Basic HDFS commands
3. Write commands for data transfer to/from HDFS
4. Visualising the HDFS file system using HUE

**Experiment 3: Importing data using Sqoop**

1. Login to MySQL
2. Create a database **upes\_db**
3. Create table **students** in the **upes\_db** database

|  |  |  |  |
| --- | --- | --- | --- |
| **Rollno** | **Name** | **Marks** | **Subject** |
| 1 | Amit | 90 | Physics |
| 2 | Sumit | 85 | Math |
| 3 | Ram | 88 | Computer |
| 4 | Sandeep | 88 | Physics |
| 5 | John | 86 | Math |
| 6 | Tom | 85 | Computer |
| 7 | Mani | 89 | Physics |
| 8 | Akshay | 88 | Math |
| 9 | Arun | 91 | Computer |

1. Using Sqoop transfer all the data of **students** table into HDFS.
2. Using Sqoop transfer all the data of **students** table into HDFS to a specific folder

**students\_1**.

1. Using Sqoop transfer all the data of **students** table who are studying **Physics**

subject into HDFS to a specific folder **students\_2.**

1. Using Sqoop transfer all the data of **students** table who are studying **Computer**

subject into HDFS to a specific folder **students\_pass,** without specifying the

password on the command line**.**

1. Import **students** table from MySQL as a text file to the destination folder

**students\_text.** Fields should be terminated by a tab character ("\t") character and

lines should be terminated by newline character ("\n").

**Experiment 4: Compression and Data Format data using Sqoop**

1. Import **students** table from MySQL into HDFS to the destination folder

**students\_avro**. The file should be stored as Avro file.

1. Import **students** table from MySQL into HDFS to the destination folder

**students\_parquet**. The file should be stored as parquet file.

1. Import **students** table from MySQL into HDFS to the destination folder

**students\_sequence**. The file should be stored as a sequence file.

1. Import **students** table from MySQL into HDFS to the destination folder

**students\_compress**. Decrease the size occupied by the generated file using **GZip**

codec.

1. Import **students** table from MySQL into HDFS to the destination folder

**students\_mappers**. Create a number of mapper 2 and use **snappy**

compression.

1. Create a new table **library** in the **upes\_db** database and import all tables from

MySQL database **upes\_db** into HDFS as Avro data files use compression.

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Author** |
| 1 | Java Programming | James Gosling |
| 2 | DBMS | Navathe |
| 3 | C Language | Yashwant Kanetkar |
| 4 | Big Data | Tom |
| 5 | Statistics | Atul |
| 6 | Networking | William |

**Experiment 5:** Incremental import using Sqoop

1. In Students table insert more data as below

|  |  |  |  |
| --- | --- | --- | --- |
| **Rollno** | **Name** | **Marks** | **Subject** |
| 10 | Akshat | 88 | Physics |
| 11 | Akshay | 85 | Math |
| 12 | Sunil | 87 | Computer |
| 13 | Mandeep | 84 | Physics |

1. Using Sqoop transfer only new rows of **students** table into HDFS to a specific

folder **students\_increment** and verify the content at the target directory in HDFS.

1. Remove primary key from **Library** table in the upes\_db database using (alter,

drop, modify) SQL commands.

1. Using Sqoop transfer all the data of the **Library** table into HDFS to a specific folder

**library\_noPk** without using primary key and verify the content at the target

directory in HDFS.

1. Using Sqoop do the following. Read the entire steps before you create a Sqoop

job.

1. Create a Sqoop job Import Students table as the text file to directory **Students\_job**.
2. Import all the new inserted three records to **Students** table from MySQL.
3. Run the Sqoop job so that only newly added records can be pulled from MySQL.
4. Validate to make sure that no duplicate records in **HDFS**

**Experiment 6:** Free form query, export and Sqoop merger

1. Create a MySQL table named **students\_1** and load data from **/home/cloudera/students\_1**. Validate to make sure the records have been added to the database.
2. Using Sqoop, import **students\_replica** table from MYSQL into HDFS such that fields are separated by a '|' and lines are separated by '\n'. Null values are represented as -1 for numbers and "NOT-AVAILABLE" for strings. Only records with roll no greater than or equal to 1 and less than or equal to 88 should be imported and use 3 mappers for importing. The destination file should be stored as a text file to directory **/home/cloudera/students\_replica**
3. Using Sqoop transfer all the data of **students** table who are studying **Physics** subject into HDFS to a specific folder **students\_query,** use SQL query to import the data.
4. Using Sqoop transfer the data of **students** table into HDFS to a specific folder **students\_m1** only student roll no less than or equal to 5, use SQL query to import the data.
5. Using Sqoop transfer all the data of **students** table into HDFS to a specific folder **students\_m2** only student roll no greater than 5.
6. Using sqoop merge data available in **students\_m1** and **students\_m2** to produce a new set of files in **students\_both.**

**Experiment 7:** Lab setup and configuration for Kakfa

1. Install JDK 8 or higher (it’s a pre-requisite)
2. Download Kafka, install and configure the properties file for Kafka and

Zookeeper.

1. Installing Java 8 & IntelliJ Community Edition.

**Experiment 8 and 9:** Kafka Producer and Consumer - Command Line

1. Create a topic name first\_topic with 1-replication factor and one partition.
2. Verify if the topic is created or not.
3. Describe topic first\_topic
4. Create a topic name second\_topic with 3-replication factor and one partition ( use

configuration properties file).

1. Verify all the topics
2. Describe topic second\_topic to check all ISR.
3. Send message to the first\_topic.
4. In new window read the first\_topic.
5. Read the first\_topic from the beginning.
6. Modify the first\_topic change partition to 3.
7. Delete the second\_topic

**Experiment 10:** Kafka Producer and Consumer – Java application

1. Create a Producer application using Java to implement fire and forget method of

sending messages in Kafka.

1. Create the consumer application to read the message from the Producer.
2. Create a Producer application using Java to implement Synchronous way of

sending messages in Kafka.

1. Create the consumer application to read the message from the Producer.
2. Create a Producer application using Java to implement Asynchronous way of

sending messages in Kafka.

1. Create the consumer application or through console window to read the message

from the Producer.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Kathleen Ting, and Jarek Jarcec Cecho, “Apache Sqoop Cookbook”, O'Reilly Media, 2013. 2. Neha Narkhede, Gwen Shapira, and Todd Palino, “Kafka: The Definitive Guide”, O'Reilly Media, 2017. |
| **Reference books** | 1. Sqoop: O'Reilly Media 2. The Big Data Ingestion: O'Reilly Media |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Wireless Sensor Networks** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Computer Networks** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the fundamentals of next generation sensor network platforms and applications including cyber-physical systems, healthcare, smart city and Internet of Things (IoT).

2. To design of energy efficient MAC protocols for Wireless Sensor Networks in IoT environment

3. To development of routing and fusion algorithms for efficient dissemination of sensor data and service discovery in next generation networks.

4. To hands-on-experience to sensor network simulators.

**Course Outcomes**

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

**CO1.** Discuss basic sensor network concepts.

**CO2.** Analyse physical layer issues and Medium Access Control Protocols.

**CO3.** Discuss network and transport layer characteristics and protocols and conventional protocols.

**CO4.** Describe the application of IOT in industrial automation and identify real world design constraints.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **-** | **1** | **-** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** |
| **CO 2** | **2** | **-** | **1** | **-** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** |
| **CO 3** | **2** | **-** | **1** | **-** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** |
| **CO4** | **2** | **-** | **1** | **-** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** |
| **Average** | **2** | **-** | **1** | **-** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I 13 lecture hours**

**INTRODUCTION & CHARACTERISTICS OF WSN**

Hardware Components, Energy Consumption of Sensor Nodes, Operating Systems and execution environments, Network Architecture, Comparison of WSNs with Ad-hoc Networks, Wireless Communication Basics (Radio Waves, Modulation, Encoding, etc.)

**Unit II 8 lecture hours**

**PHYSICAL LAYER**

Frequency Allocation, Packet transmission and Synchronization

**Unit III 12 lecture hours**

**MAC AND LINK LAYER PROTOCOLS**

IEEE 802.15.4 MAC Protocol, Contention based protocols, Schedule based protocols, Error Control, Link Management

**Unit IV 15 lecture hours**

**ROUTING PROTOCOLS AND DATA/ CONTENT CENTRIC NETWORKING**

Energy-efficient unicast, Broadcast and multicast, Geographic Routing, Data-Centric routing, Data Aggregation, Flat Routing (Flooding, Gossiping), Hierarchical Routing (LEACH, PEGASIS), Location-based Routing (GPSR, GEAR)

**Unit V 12 lecture hours**

**APPLICATIONS OF WSNs AND FUTURE TRENDS**

IPv6, CoAP, Building Automation, Internet of Things, Smart Agriculture, Perimeter monitoring, Object Tracking.

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| Textbooks | 1. Wireless Sensor Networks: Principles, Design, and Applications" – K. Sohraby, D. Minoli, T. Znati 2. Protocols and Architectures for Wireless Sensor Networks" – Holger Karl & Andreas Willig |
| Reference books | 1. Waltenegus Dargie, Christian Poellabauer , “Fundamentals of Wireless Sensor Networks, Theory and Practice”, Wiley Series on wireless Communication and Mobile Computing, 2011 2. Kazem Sohraby, Daniel manoli , “Wireless Sensor networks- Technology, Protocols and Applications”, Wiley InterScience Publications 2010. 3. Bhaskar Krishnamachari , “ Networking Wireless Sensors”, Cambridge University Press, 2005 |
| Web Resources |  |
| Journals |  |
| MOOCs, online courses |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Wireless Sensor Networks Lab**

**List of Experiments**

1. Introduction to WSN Simulation Tools

* NS-2, NS-3, Contiki Cooja, TOSSIM

1. Programming Wireless Sensor Nodes

* Writing and Executing a Simple Code for Node Communication

1. Implementation of Data Transmission in IEEE 802.15.4
2. Simulation of Different MAC Protocols (CSMA, TDMA, S-MAC)
3. Energy Consumption Analysis in Sensor Nodes
4. Routing Protocol Implementation

* Implementing Flooding and Gossiping Protocols

1. Simulation of Data-Centric Routing Protocols (SPIN, Directed Diffusion)
2. Network Topology Formation and Visualization in WSNs
3. Implementation of Localization Algorithms in WSN
4. Implementing a Smart Agriculture Use Case with WSN Simulation
5. Security Attack Simulation in WSN

* Analyzing Data Tampering and Malicious Node Behavior

1. IoT Integration with WSN

* Using MQTT and CoAP Protocols for Sensor Data Communication

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Wireless Sensor Networks: Principles, Design, and Applications" – K. Sohraby, D. Minoli, T. Znati 2. Protocols and Architectures for Wireless Sensor Networks" – Holger Karl & Andreas Willig |
| **Reference books** | 1. Waltenegus Dargie, Christian Poellabauer , “Fundamentals of Wireless Sensor Networks, Theory and Practice”, Wiley Series on wireless Communication and Mobile Computing, 2011 2. Kazem Sohraby, Daniel manoli , “Wireless Sensor networks- Technology, Protocols and Applications”, Wiley InterScience Publications 2010.   Bhaskar Krishnamachari , “ Networking Wireless Sensors”, Cambridge University Press, 2005 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Oparation Research and Game Theorey** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 90** | | | | |
| **Prerequisite(s):** | **Mathematics, Algorithm** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

To classify and formulate real-life problem for modelling, solving and applying for decision making.

* To study the formulation and various methods of solutions for linear programming, transportation, assignment , CPM and PERT problems
* To solve  problems using  dynamic programming method

**Course Outcomes**

After completion of course, students would be able to:

1. Analyse problems in engineering, management, or business environment, focusing on important details

2. Formulate of real problems in terms of input-output-parameters relationships and identify the solution procedure.

3. Apply optimization techniques in real time problem solving.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **2** | **3** | **3** | **3** | **-** | **2** | **-** | **3** | **4** | **3** | **2** | **2** | **2** | **3** |
| **CO 2** | **4** | **4** | **4** | **-** | **4** | **4** | **-** | **4** | **3** | **3** | **3** | **2** | **2** | **2** | **3** |
| **CO 3** | **3** | **3** | **3** | **2** | **2** | **2** | **2** | **2** | **-** | **2** | **2** | **1** | **1** | **1** | **2** |
| **Average** | **3** | **3** | **3.3** | **2.5** | **3** | **3** | **2** | **3** | **3** | **3** | **3** | **3.3** | **2.5** | **3** | **3** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: 20 Lecture Hours**

Introduction to OR Models, Complex OR Models, Graphical Method for LPP, Convex sets, Simplex Method. Big M Method, Two Phase, Multiple solutions of LPP, Unbounded solution of LPP, Infeasible solution of LPP. Revised Simplex Method, Case studies and Exercises-I, Case studies and Exercises-II, Case studies and Exercises - III, Primal Dual Construction.

Unit II: 20 Lecture Hours

Weak Duality Theorem, More variant of Duality Theorems, Primal-Dual relationship of solutions, Dual Simplex Method, Sensitivity Analysis-I. Sensitivity Analysis-II, Case studies and Exercises - I, Case studies and Exercises - II, Integer Programming Goal Programming, Multi-Objective Programming, Dynamic Programming, Transportation Problem, Assignment Problem, Case studies and,Exercises. Processing n Jobs on Two Machines, Processing n Jobs through Three Machines, Processing two jobs through m machines, Processing n jobs through m machines, Case studies and Exercises.

Unit III: 20 Lecture Hours

Two Person Zero-Sum Game, Theorems of Game Theory, Solution of Mixed Strategy Games, Linear Programming method for solving games, Case studies and Exercises. Sensitivity Analysis: Changes in Objective Function, Sensitivity Analysis: Changes in RHS, The Transportation Model Basic Assumptions. easible Solution: The Northwest Method, The Lowest Cost Method Optimal Solution: The Stepping Stone Method, Modified Distribution (MODI) Method.

List of experiments

1. Simplex method using given data.

2. Integer Programming

3. Goal Programming,

4. Multi-Objective Programming,

5. Dynamic Programming,

6. Transportation Problem,

7. Assignment Problem,

8. Two Person Zero-Sum Game,

9. Solution of Mixed Strategy Games,

10. Linear Programming method for solving games,

11. The Stepping Stone Method,

12. Modified Distribution (MODI) Method.

30 Lab Hours

Refrences :

|  |  |
| --- | --- |
| **Textbooks** | 1. Mittal, K. V. and Mohan, C. “Optimization Methods in Operations Research and Systems Analysis”, New Age, 2003.   aha, H.A. : “Operations Research – An Introduction”, Prentice Hall, (7th Edition), 2002. |
| **Reference books** | 1. Ravindran, A. , Phillips, D. T and Solberg, J. J. “Operations Research: Principles and Practice”, John Willey and Sons, 2nd Edition, 2009. |
| **Web Resources** |  |
| **Journals** | Journals from IEEE, SAAM, ACM, Elsevier. |
| **MOOCs, online courses** | MIT open courseware, NPTEL, Standord. |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Parallel Computing** | | **4** | **0** | **1** | **5** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 90** | | | | | |
| **Prerequisite(s):** | **Data Structures and algorithms and Programming in C** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. Provide a complete understanding of parallel computing concepts, architectures, and execution models.
2. Introduce performance analysis techniques for parallel applications.
3. To develop proficiency in parallel programming paradigms.
4. Familiarize students with distributed and cloud-based computing models

**Course Outcomes**

After completion of course, students would be able to:

1. Understand different types of parallel architectures and memory models.
2. Develop parallel applications using different programming models like OpenMP, MPI and CUDA.
3. Apply parallel computing techniques to sorting, matrix operations and graph algorithms.
4. Understand the concepts of distributed and cloud based parallel systems.
5. Demonstrate the implementation of parallel programming models such as OpenMP, MPI, and CUDA for high-performance computing applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO 12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **1** | **2** | **-** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **-** | **1** |
| **CO 2** | **1** | **1** | **-** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **2** | **-** |
| **CO 3** | **1** | **2** | **-** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **2** | **-** |
| **CO 4** | **2** | **1** | **-** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **1** | **-** |
| **CO 5** | **1** | **1** | **-** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **3** | **3** | **2** |
| **Average** | **1** | **1.4** | **-** | **-** | **2** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **2.8** | **1.6** | **0.6** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: 10 Lecture Hours**

Evolution of parallel computing, Need for parallelism in modern computing, Types of parallelism - Task parallelism vs. Data parallelism, Performance metrics - Speedup, scalability, efficiency, Amdahl’s Law, and Gustafson’s Law.

Flynn’s Taxonomy (SISD, SIMD, MISD, MIMD), Shared Memory vs. Distributed Memory Architectures

**Unit II: 12 Lecture Hours**

Shared Memory Programming - Threads, POSIX Threads (Pthreads), OpenMP – pragmas and directives, Scope of variables, reduction clause, loop carried dependency, scheduling.

Message Passing – MPI (Message Passing Interface), Point-to-Point communication, communication patterns - synchronous and asynchronous communication, collective communication - gather, scatter, broadcast, reduce, barrier. Fault tolerance in MPI, Checkpoint and recovery, handling node failures, combining MPI and OpenMP.

**Unit III: 12 Lecture Hours**

CUDA programming model, Introduction to GPU architectures, Thread hierarchy – grids, blocks. Memory Hierarchy – global, shared, constant, texture and registers. Writing CUDA kernels – Syntax and best practices, CUDA trapezoidal rule, Using CUDA streams for computation with data transfers. Optimizing GPU applications.

**Unit IV: 14 Lecture Hours**

Designing parallel algorithms – parallel algorithm design principles, divide and conquer strategies for parallelism, graph algorithms, matrix operations and numerical simulations. Parallel sorting algorithms – parallel merge sort, quick sort. Load balancing – static and dynamic scheduling strategies.

**Unit V: 12 Lecture Hours**

Parallel computing in distributed systems, role of MPI in distributed environments, Overview of Hadoop for processing large datasets in parallel, MapReduce programming model and Hadoop - design, implementation, and performance considerations. Introduction to cloud based parallel computing – AWS, Google Cloud and Azure, Introduction to edge and fog computing in parallel systems

**Total lecture Hours 60**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. A Grama, A Gupta, G Karypis, and V Kumar, Introduction to Parallel Computing. 2nd Ed., Addison-Wesley, 2003. 2. Peter S Pacheco, Matthew Malensek, An Introduction to Parallel Programming, second edition, Morgan Kauffman, 2021. |
| **Reference books** | 1. Michael J. Quinn, Parallel Computing. Theory and Practice, McGraw Hill Education; 2nd edition, 2017. 2. Calvin Lin, Lawrence Snyder – Principles of Parallel Programming, Pearson, Ist Edition, 2008 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** | 1. https://nptel.ac.in/courses/106102114 2. <https://nptel.ac.in/courses/106102163> |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Parallel Computing Lab**

**List of Experiments**

**Experiment 1**

Write an OpenMP program that prints "Hello, World!" from multiple threads and determine the number of threads.

Write a simple loop-based computation program in C and parallelize it using OpenMP and compare parallel and sequential execution times.

**Experiment 2**

Write a OpenMP program to calculate n Fibonacci numbers using tasks.

Write a sequential matrix multiplication program using OpenMP.

**Experiment 3**

**Write a program using OpenMP program to sum an array using reduction(+:sum).**

**Write a program to implement BFS using OpenMP and compare it with the sequential BFS traversal time.**

**Experiment 4:**

**Write an MPI program where each process prints "Hello from processes".**

**Write a program to implement Merge Sort using MPI.**

**Experiment 5:**

**Write a program to send an integer from one process to another using MPI\_Send and MPI\_Recv.**

**Rewrite the above program for multiple processes using a ring communication pattern.**

**Experiment 6:**

**Write a program using MPI\_Bcast to send a value to all the other processes.**

**Write a MPI program to show how a process gather a value from all other processes using MPI\_Gather.**

**Experiment 7:**

**Write a CUDA program where a kernel function prints "Hello from CUDA".**

**Write a program to write a kernel for sum reduction.**

**Experiment 8:**

**Write a CUDA program to perform element-wise addition of two large vectors.**

**Compute the dot product of two vectors using CUDA parallel reduction.**

**Experiment 9:**

**Install Hadoop and setup a single-node and multi-node Hadoop cluster.**

**Implement and run the classic WordCount example on a large text dataset and analyse the map and reduce phases.**

**Experiment 10:**

**Deploy a Hadoop cluster on AWS and execute the wordcount MapReduce job.**

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | A Grama, A Gupta, G Karypis, and V Kumar, Introduction to Parallel Computing. 2nd Ed., Addison-Wesley, 2003. |
| **Reference books** | Thomas la Cour Jansen, Basic Parallel Programming with OpenMP: A guide to cutting your scientific calculations in smaller pieces. TLC Publishing.  Brian Tuomanen, Hands-On GPU Programming with Python and CUDA (2018) Packt Publishing, ISBN-10 ‏ : ‎ 1788993918 |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** | https://nptel.ac.in/courses/106102114  <https://nptel.ac.in/courses/106102163> |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **DevOps Overview** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To enable learners realize various aspects of DevOps Ecosystem and agile led development.
2. To give the students a perspective to grasp the need for Minimum viable product led development using Sprints
3. To enable learners to pick up fundamentals of Continuous Integration and Continuous Deployment.
4. To enable students, acquire thorough understanding of difference between version control system and distributed version control system.

**Course Outcomes**

After completion of course, students would be able to:

CO1 Explain and compare traditional software development methodologies like waterfall

with agile development.

CO2. Prepare quick MVP prototypes for modules and functionalities.

CO3. Explain fundamentals of Continuous Integration and Continuous Delivery (CICD)

pipeline.

CO4. Analyze workflows in various version control system like Git, Sun and Mercurial.

CO5. Implement core DevOps principles, including continuous integration, continuous

deployment, automation, and monitoring, to enhance software development efficiency

and operational reliability in real-world scenarios.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Outcomes** | **PO**  **1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO**  **9** | **PO**  **10** | **PO**  **11** | **PO**  **12** | **PSO**  **1** | **PSO**  **2** | **PSO**  **3** |
| **CO1** | **2** | **-** | **1** | **-** | **1** | **-** | **-** | **1** | **1** | **-** | **-** | **-** | **-** | **2** | **-** |
| **CO2** | **2** | **2** | **2** | **-** | **3** | **2** | **-** | **3** | **3** | **2** | **-** | **2** | **1** | **2** | **-** |
| **CO3** | **2** | **1** | **3** | **-** | **3** | **2** | **-** | **3** | **3** | **2** | **-** | **2** | **-** | **1** | **-** |
| **CO4** | **2** | **1** | **-** | **-** | **2** | **-** | **-** | **1** | **2** | **-** | **-** | **-** | **-** | **-** | **-** |
| **CO5** | **2** | **-** | **2** | **-** | **-** | **1** | **-** | **-** | **-** | **-** | **-** | **-** | **1** | **2** | **1** |
| **Average** | **2** | **0.8** | **1.6** | **-** | **1.8** | **1** | **-** | **1.6** | **1.8** | **0.8** | **-** | **0.8** | **0.4** | **1.4** | **0.2** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**UNIT I 12 Lecture Hours**

**Traditional Software Development and Rise of Agile Methodologies**

The Advent of Software Engineering, Waterfall method, Developers vs IT Operations conflict

Agile movement in 2000, Agile Vs Waterfall Method, Iterative Agile Software Development, Individual and team interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan

**UNIT II 10 Lecture Hours**

**Definition and Purpose of DevOps**

Introduction to DevOps, DevOps and Agile.Minimum Viable Product, Application Deployment, Continuous Integration, Continuous Delivery

CAMS – Culture, CAMS – Automation, CAMS – Measurement, CAMS – Sharing, Test-Driven Development, Configuration Management, Infrastructure Automation, Root Cause Analysis, Blamelessness, Organizational Learning

**UNIT III: 8 Lecture Hours**

**Typical Toolkit for DevOps, Source Code Management History and Overview**

Introduction to continuous integration and deployment, Version control system

Examples - SVN, Mercury and Git, History - Linux and Git by Linus Torvalds

**UNIT IV: 8 Lecture Hours**

**Version Control System Vs Distributed Version Control System**

Local repository, Advantages of distributed version control system, The Multiple Repositories Model, Completely resetting local environment, Revert - cancelling out changes

**Unit V: 7 Lecture Hours**

**Monitoring and Security**

Importance of monitoring in DevOps, implementing application and infrastructure monitoring, Log aggregation and analysis, Security and Compliance, DevOps security principles and practices, Implementing security controls in CI/CD pipelines, Compliance considerations in DevOps.

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. The DevOps Handbook - Book by Gene Kim, Jez Humble, Patrick Debois, and Willis Willis |
| **Reference books** | 1. What is DevOps? - by Mike Loukides 2. Pro Git – Book by Scott Chacon and Ben Straub (available at https://git-scm.com/book/). |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**DevOps Overview Lab**

**List of Experiments**

1. **Introduction to Version Control with Git & GitHub**
   * Setting up Git, creating repositories, branching, merging, and handling pull requests.
2. **Continuous Integration (CI) with Jenkins**
   * Installing Jenkins, setting up a basic CI pipeline, and automating builds.
3. **Containerization with Docker**
   * Building Docker images, running containers, and managing Docker networks & volumes.
4. **Container Orchestration with Kubernetes**
   * Deploying and managing applications in Kubernetes using Pods, Deployments, and Services.
5. **Infrastructure as Code (IaC) with Terraform**
   * Writing Terraform scripts to provision cloud infrastructure on AWS/Azure/GCP.
6. **Configuration Management with Ansible**
   * Automating software installation and configuration using Ansible playbooks.
7. **Monitoring and Logging with Prometheus & Grafana**
   * Setting up Prometheus for metrics collection and visualizing data in Grafana.
8. **Automated Testing in DevOps Pipelines**
   * Implementing unit testing and integration testing in CI/CD workflows.
9. **Continuous Deployment (CD) with GitHub Actions**
   * Automating application deployment to cloud servers using GitHub Actions.
10. **Security in DevOps (DevSecOps)**

* Implementing security checks in CI/CD pipelines using tools like SonarQube or Trivy.

**Total Lab hours 30**

|  |  |
| --- | --- |
| **Textbooks** |  |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Distributed Computing** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the basic underlying concepts of a distributed system.

2. To comprehend the concept of Clock Synchronization

3. To explore mutual exclusion algorithms and deadlock handling in the distributed system

4. To implement different distributed computing concepts through programing.

**Course Outcomes**

After completion of course, students would be able to:

1. Explain the techniques for clock synchronization.
2. Describe mutual exclusion algorithms and distributed deadlock handling.
3. Comprehend the techniques of distributed deadlock handling.
4. Understand the concepts of rollback and recovery.
5. Implement, test and debug different distributed computing concepts like, clock synchronization, election algorithms, mutual exclusion, distributed deadlock handling etc.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **1** | **1** | **1** | **1** | **1** | **1** | **2** | **1** | **3** | **1** | **2** | **-** | **1** | **2** |
| **CO 2** | **2** | **1** | **1** | **1** | **1** | **1** | **1** | **2** | **1** | **3** | **1** | **2** | **-** | **1** | **2** |
| **CO 3** | **1** | **3** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **2** | **1** | **2** | **-** | **1** | **2** |
| **CO 4** | **1** | **2** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **2** | **1** | **2** | **-** | **1** | **2** |
| **CO 5** | **3** | **2** | **3** | **2** | **1** | **1** | **1** | **1** | **2** | **2** | **3** | **2** | **3** | **3** | **2** |
| **Average** | **1.8** | **1.8** | **1.4** | **1.2** | **1** | **1** | **1** | **1.4** | **1.2** | **2.4** | **1.4** | **2** | **.6** | **1.4** | **2** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction 10 Lecture Hours**

Introduction: Definition-Relation to Computer System Components – Motivation – Message Passing ystems versus Shared Memory Systems – 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.

**Unit II: LOGICAL TIME AND GLOBAL STATE 10 Lecture Hours**

Logical Time: Physical Clock Synchronization: NTP – A Framework for a System of Logical Clocks – Scalar Time – Vector Time; Message Ordering and Group Communication: Message Ordering Paradigms – Asynchronous Execution with Synchronous Communication – Synchronous Program Order on Asynchronous System – Group Communication – Causal Order – Total Order; Global State and Snapshot Recording Algorithms: Introduction – System Model and Definitions – Snapshot Algorithms for FIFO Channels.

**Unit III: DISTRIBUTED MUTEX AND DEADLOCK 10 Lecture Hours**

Distributed Mutual exclusion Algorithms: Introduction – Preliminaries – Lamport’s algorithm – RicartAgrawala’s Algorithm –– Token-Based Algorithms – Suzuki-Kasami’s Broadcast Algorithm; Deadlock Detection in Distributed Systems: Introduction – System Model – Preliminaries – Models of Deadlocks – Chandy-Misra-Haas Algorithm for the AND model and OR Model.

**Unit IV: CONSENSUS AND RECOVERY 10 Lecture Hours**

Consensus and Agreement Algorithms: Problem Definition – Overview of Results – Agreement in a Failure-Free System(Synchronous and Asynchronous) – Agreement in Synchronous Systems with Failures; Checkpointing and Rollback Recovery: Introduction – Background and Definitions – Issues in Failure Recovery – Checkpoint-based Recovery – Coordinated Checkpointing Algorithm – – Algorithm for Asynchronous Check pointing and Recovery

**Unit V: CLOUD COMPUTING 5 Lecture Hours**

Definition of Cloud Computing – Characteristics of Cloud – Cloud Deployment Models – Cloud Service Models – Driving Factors and Challenges of Cloud – Virtualization – Load Balancing – Scalability and Elasticity – Replication – Monitoring – Cloud Services and Platforms: Compute Services – Storage Services – Application Services

**Total Lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Kshemkalyani Ajay D, Mukesh Singhal, “Distributed Computing: Principles, Algorithms and Systems”, Cambridge Press, 2011.  2. Mukesh Singhal, Niranjan G Shivaratri, “Advanced Concepts in Operating systems”, McGraw Hill Publishers, 1994. |
| **Reference books** | 1. George Coulouris, Jean Dollimore, Time Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education, 2012.  2. Pradeep L Sinha, “Distributed Operating Systems: Concepts and Design”, Prentice Hall of India, 2007.  3. Tanenbaum A S, Van Steen M, “Distributed Systems: Principles and Paradigms”, Pearson Education, 2007.  4. Liu M L, “Distributed Computing: Principles and Applications”, Pearson Education, 2004.  5. Nancy A Lynch, “Distributed Algorithms”, Morgan Kaufman Publishers, 2003.  6. Arshdeep Bagga, Vijay Madisetti, “ Cloud Computing: A Hands-On Approach”, Universities Press, 2014. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

****Distributed Computing Lab****

1. **Inter-process communication**
2. **Client/Server using RPC/RMI**
3. **Group Communication**
4. **Clock Synchronization algorithms**
5. **Election Algorithm**
6. **Mutual Exclusion Algorithm**
7. **Deadlock Management in Distributed System**
8. **Load Balancing 9 Distributed shared Memory**
9. **Distributed File System (AFS/CODA)**
10. **Case Study: CORBA 12 Case Study: Android Stack**

****Total Lab Hours 30****

# References\*

|  |  |
| --- | --- |
| **Textbooks** | 1. Kshemkalyani Ajay D, Mukesh Singhal, “Distributed Computing: Principles, Algorithms and Systems”, Cambridge Press, 2011.   2. Mukesh Singhal, Niranjan G Shivaratri, “Advanced Concepts in Operating systems”, McGraw Hill Publishers, 1994. |
| **Reference books** | 1. George Coulouris, Jean Dollimore, Time Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education, 2012.  2. Pradeep L Sinha, “Distributed Operating Systems: Concepts and Design”, Prentice Hall of India, 2007.  3. Tanenbaum A S, Van Steen M, “Distributed Systems: Principles and Paradigms”, Pearson Education, 2007.  4. Liu M L, “Distributed Computing: Principles and Applications”, Pearson Education, 2004. |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code XXX** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Mobile Computing** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 05** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | **Data Communication and Networks** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To Understand the basic premises of Mobile Computing
2. To comprehend cellular systems.
3. To learn about different mobile layers.
4. To install software like NS2/NS3.
5. To write TCL script for simulating different routing algorithms.
6. To design and develop basic mobile applications.

**Course Outcomes**

After completion of course, students would be able to:

1. Describe the importance of mobile computing.
2. Describe the working principles of cellular networks.
3. Understand the importance and requirements of mobile network layer.
4. Understand the importance and requirements of mobile transport layer.
5. Implement different mobile routing algorithms and develop mobile friendly applications.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **2** | **1** | **1** | **1** | **-** | **-** | **-** | **1** | **3** | **1** | **2** | **-** | **-** | **1** |
| **CO 2** | **2** | **2** | **1** | **1** | **1** | **-** | **-** | **-** | **1** | **3** | **1** | **2** | **-** | **-** | **1** |
| **CO 3** | **2** | **2** | **2** | **2** | **1** | **-** | **-** | **-** | **1** | **2** | **1** | **2** | **-** | **-** | **1** |
| **CO 4** | **2** | **2** | **2** | **2** | **1** | **-** | **-** | **-** | **1** | **2** | **1** | **2** | **-** | **-** | **1** |
| **CO 5** | **3** | **3** | **3** | **3** | **3** | **1** | **2** | **3** | **3** | **2** | **3** | **3** | **3** | **3** | **2** |
| **Average** | **2.2** | **2.2** | **1.8** | **1.8** | **1.4** | **.2** | **.4** | **.6** | **1.4** | **2.4** | **1.4** | **2.2** | **.6** | **.6** | **1.2** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: Introduction 5 Lecture Hours**

Introduction to Mobile Computing — Applications of Mobile Computing- Generations of Mobile Communication Technologies- Multiplexing — Spread spectrum -MAC Protocols — SDMA- TDMA- FDMA- CDMA

### **Unit II: MOBILE TELECOMMUNICATION SYSTEM** **10 Lecture Hours**

Introduction to Cellular Systems — GSM — Services & Architecture — Protocols — Connection Establishment — Frequency Allocation — Routing — Mobility Management — Security — GPRS- UMTS — Architecture — Handover — Security

### **Unit III: MOBILE NETWORK LAYER 10 Lecture Hours**

Mobile IP — DHCP — AdHoc– Proactive protocol-DSDV, Reactive Routing Protocols — DSR, AODV , Hybrid routing –ZRP, Multicast Routing- ODMRP, Vehicular Ad Hoc networks ( VANET) –MANET Vs VANET — Security.

### **Unit IV: MOBILE TRANSPORT AND APPLICATION LAYER**

**10 Lecture Hours**

Mobile TCP– WAP — Architecture — WDP — WTLS — WTP –WSP — WAE — WTA Architecture — WML

### **Unit V: MOBILE PLATFORMS AND APPLICATIONS**

**10 Lecture Hours**

Mobile Device Operating Systems — Special Constraints & Requirements — Commercial Mobile Operating Systems — Software Development Kit: iOS, Android, BlackBerry, Windows Phone — MCommerce — Structure — Pros & Cons — Mobile Payment System — Security Issues

**Total lecture Hours 45**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | * 1. Mobile Computing: Theory and P{ractice, Kumkum Garg. Pearson Education India, 2010.  2. Principles of Mobile Computing and Communications By: [Mazliza Othman](https://www.cb-india.com/mazliza-othman/) | Publisher:  [CRC Press](https://www.cb-india.com/index.php?match=all&pcode_from_q=Y&pshort=N&pfull=N&pname=Y&pkeywords=Y&search_performed=Y&q=&dispatch=products.search&features_hash=11-278) |
| **Reference books** | 1. FUNDAMENTALS OF MOBILE COMPUTING[PATTNAIK, PRASANT KUMAR](https://www.phindia.com/Books/Author/OTc4ODEyMDM1MTgxMw)[MALL, RAJIB](https://www.phindia.com/Books/Author/OTc4ODEyMDM1MTgxMw) Second Edition |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Mobile computing Lab**

**List of Experiments**

1. Installing NS2 or NS3.

2) Create a TCL Script for the following network.

N2 1MB, 10ms, DropTail N1

Create FTP traffic over TCP. Find out the throughput using GREP

command.

3) Create a TCL Script for the following network.

N2 1MB, 10ms, DropTail N1

Create CBR traffic over UDP. Find out the throughput using GREP

command.

4) Simulate the Distance Vector Routing Algorithm and Analyze the

performance metrics such as throughput, packet drop rate etc

5) Simulate the Link State Routing Algorithm and Analyze the

performance metrics such as throughput, packet drop rate etc

6) Develop a simple mobile application for swapping images by using

either Android or IBMWorklight.

7) Develop a mobile calculator application that performs addition,

subtraction, multiplication, division, modulus operations on mobile

by using either Android or IBM Worklight.

8) Design and develop a mobile application to validate user name and

password by using either Android or IBM worklight.

9) Design and develop a College Information system by using either

Android or IBM worklight.

10) Simulate the mobile chatting application by using either Android or

IBMWorklight.

**Total Lab hours 30**

# References\*

|  |  |
| --- | --- |
| **Textbooks** | Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps Paperback – 29 September 2009 [Brian Fling](https://www.amazon.in/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=Brian+Fling&search-alias=stripbooks), O’Reilly |
| **Reference books** |  |
| **Web Resources** |  |
| **Journals** |  |
| **MOOCs, online courses** |  |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Fog and Edge Networks** | | **3** | **0** | **1** | **4** |
| **Total Units to be Covered: 06** | | **Total Contact Hours: 75** | | | | |
| **Prerequisite(s):** | **Cloud Computing, Data Communication and Networks** | | **Syllabus version: 1.0** | | | |

****Course Objectives****

1. To understand the major components of Fog and Edge Networks architectures such as middleware, interaction protocols, network management and autonomic service management.
2. To understand the data collection analysis and learning and do analytics at the edge of modern networks.
3. Improve performance at the fog and edge networks and analyze the latest deployment systems and platforms to design application

**Course Outcomes**

After completion of course, students would be able to:

1. Understand the foundations of fog and edge computing networks and different architectures.
2. Design fog and edge networks based systems and applications using reference architectures.
3. Understand and apply data collection, analysis, decision making and learning methodologies over the edge for different applications
4. Apply optimization techniques for edge and fog computing

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  **Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO 11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **2** | **2** | **3** | **3** | **3** | **-** | **2** | **-** | **3** | **4** | **2** | **3** | **2** | **2** | **3** |
| **CO 2** | **4** | **4** | **4** | **-** | **4** | **4** | **-** | **4** | **3** | **3** | **3** | **2** | **2** | **2** | **3** |
| **CO 3** | **3** | **3** | **3** | **2** | **2** | **2** | **2** | **2** | **-** | **2** | **-** | **1** | **1** | **1** | **2** |
| **CO 4** | **3** | **3** | **3** | **3** | **2** | **1** | **-** | **3** | **-** | **-** | **2** | **3** | **3** | **3** | **-** |
| **Average** | **3** | **3** | **3.25** | **2.6** | **2.75** | **2.3** | **2** | **3** | **3** | **3** | **3** | **3** | **3.25** | **2.6** | **2.75** |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

**Unit I: 10 Lecture Hours**

Introduction to Fog Computing and Networks: Fog and edge Computing, Characteristics, Application Scenarios,Issues and challenges. Fog Computing Architecture: Communication and Network Model, Programming Models, Fog Architecture for smart cities, healthcare and vehicles. Fog Computing Communication Technologies: Introduction, IEEE 802.11, 4G, 5G standards, WPAN, Short-Range Technologies, LPWAN and other medium and Long-Range Technologies.

**Unit II: 15 Lecture Hours**

Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds: Introduction,Background, Network Slicing in 5G, Network Slicing in Software-Defined Clouds, Network Slicing Management in Edge and Fog, Middleware for Fog and Edge Computing, Need for Fog and Edge Computing Middleware, Clusters for Lightweight Edge Clouds, IoT Integration, Security Managementfor Edge Cloud Architectures.Fog Computing Realization for Big Data Analytics: Introduction to Big Data Analytics, Data Analytics in the Fog, Prototypes and Evaluation.

**Unit III: 15 Lecture Hours**

Fog computing requirements when applied to IoT: Scalability, Interoperability, Fog-IoT architectural model, Challenges on IoT Stack Model via TCP/IP Architecture, Data Management, filtering, Event Management, Device Management, cloudification, virualization, security and privacy issues. IntegratingIoT, Fog, Cloud Infrastructures: Methodology, Integrated C2F2T Literature by Modeling Technique byUse-Case Scenarios, Integrated C2F2T Literature by Metrics.

**Unit VI: 5 Lecture Hours**

Exploiting Fog and edge in Health Monitoring: An Architecture of a Health Monitoring IoT Based System with Fog Computing, Fog Computing Services in Smart E-Health Gateways, Discussion of Connected Components.

Fog Computing Model for Evolving Smart Transportation Applications:Introduction, Data-Driven Intelligent Transportation Systems, Fog Computing for Smart Transportation, Applications Case Study: Intelligent Traffic Lights Management (ITLM) System.

**Total lecture Hours 45**

|  |  |
| --- | --- |
| **Textbooks** | 1. Fog Computing: Theory and Practice by Assad Abbas, Samee U. Khan, Albert Y. Zomaya.  2. Fog and Edge Computing: Principles and Paradigms (Wiley Series on Parallel and DistributedComputing) by Rajkumar Buyya and Satish Narayana Srirama. |
| **Reference books** | Mir Vahid Dastjerdi and Rajkumar Buyya, ―Fog Computing: Helping the Internet of ThingsRealize its Potential, University of Melbourne. REFERENCE BOOKS:  1. Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli, ―Fog Computing and Its Role inthe Internet of Things, MCC’ 12, August 17, 2012, Helsinki, Finland. Copyright 2012 ACM 978-1-4503-1519-7/12/08... $15.00. 2. Shanhe Yi, Cheng Li, Qun Li, ―A Survey of Fog Computing: Concepts, Applications and Issues, Mobidata’ 15, ACM 978-1-4503-3524-9/15/06, DOI: 10.1145/2757384.2757397, June 21, 2015, Hangzhou, China.  3. Amir M. Rahmani, Pasi Liljeberg, Preden, Axel Jantsch, ―Fog Computing in the Internet ofThings - Intelligence at the Edge‖, Springer International Publishing, 2018.  4. Ivan Stojmenovic, Sheng Wen, “The Fog Computing Paradigm: Scenarios and Security Issues”, Proceedings, Federated Conference on Computer Science and Information Systems,pp. 1–8, 2014. |
| **Web Resources** |  |
| **Journals** | IEEETCC, IEEE Cloud computing, ACM cloud computing, JNCA, COMNET, etc. |
| **MOOCs, online courses** | MIT open courseware, NPTEL, Standord. |

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **IA** | **MID SEM** | **End Sem** | **Total** |
| Weightage (%) | 50 | 20 | 30 | 100 |

**Fog and Edge Networks Labs**

**List of Experiments**

**Experiment 1 and 2:**

**1.** Fundamental of advance communication network setup, and association.

2. 5G and other IEEE network slandered.

**Experiment 3 and 4:**

1. Cloud network deployment.
2. Cloud service model

**Experiment 5 and 7:**

1. Cloud security and network integration

**Experiment 8 and 9:**

1. **Cloud, network, fog and edge service and deployment.**

**Experiment 10 and 11:**

1. **QoS and QoE model for fog and edge.**

**Experiment 12 and 13:**

1. Fog and edge deployment in healthcare.

**Experiment 14 and 15:**

**1. Fog and edge deployment in smart IoT/cities**

**Total Lab hours 30**

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination etc.**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 % | 50 % |