Pimpri Chinchwad Education Trust's

# PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR NO. 26, PRADHIKARAN, NIGDI, PUNE 411044

(An Autonomous Institute Approved by AICTE and Affiliated to SPPU, Pune)

## Curriculum Structure and Syllabus

## of

## Third Year B. Tech. Information Technology
## (Regulations 2023)

**Effective from Academic Year 2026-27**
**(Updated with minor changes)**

## Institute Vision

To be one of the top 100 Engineering Institutes of India in coming five years by offering exemplarily Ethical, Sustainable and Value Added Quality Education through a matching ecosystem for building successful careers.

## Institute Mission

1. Serving the needs of the society at large through establishment of a state-of-art Engineering Institute.
2. Imparting right Attitude, Skills, Knowledge for self-sustenance through Quality Education.
3. Creating globally competent and Sensible engineers, researchers and entrepreneurs with an ability to think and act independently in demanding situations.

## EOMS Policy

"We at PCCOE are committed to offer exemplarily Ethical, Sustainable and Value Added Quality Education to satisfy the applicable requirements, needs and expectations of the Students and Stakeholders.

We shall strive for technical development of students by creating globally competent and sensible engineers, researchers and entrepreneurs through Quality Education.

We are committed for Institute's social responsibilities and managing Intellectual property.

We shall achieve this by establishing and strengthening state-of-the-art Engineering Institute through continual improvement in effective implementation of Educational Organizations Management Systems (EOMS)."

# Course Approval Summary

**Board of Studies - Department of Information Technology**

| Sr. No. | Name of the Course | Course Code | Page number | Signature and stamp of BoS Chairman |
|---|---|---|---|---|
| 1. | Artificial Intelligence and Machine Learning | BIT25PC01 | 2 | |
| 2. | Artificial Intelligence and Machine Learning Laboratory | BIT25PC06 | 4 | |
| 3. | Design and Analysis of Algorithms | BIT25PC07 | 6 | |
| 4. | Design and Analysis of Algorithms Laboratory | BIT25PC04 | 8 | |
| 5. | Software Engineering and Modeling Laboratory | BIT25PC05 | 12 | |
| 6. | Programme Elective Course (PEC) 1 : Fundamentals of Digital Image Processing | BIT25PE01 | 16 | |
| 7. | Programme Elective Course (PEC) 1 Laboratory : Fundamentals of Digital Image Processing Laboratory | BIT25PE04 | 18 | |
| 8. | Programme Elective Course (PEC) 1: Software Project Management | BIT25PE02 | 20 | |
| 9. | Programme Elective Course (PEC) 1 Laboratory : Software Project Management Laboratory | BIT25PE05 | 22 | |
| 10. | Programme Elective Course (PEC) 1 : Engineering a Cloud | BIT25PE03 | 24 | |
| 11. | Programme Elective Course (PEC) 1 Laboratory : Engineering a Cloud Laboratory | BIT25PE06 | 26 | |
| 12. | Operating Systems | BIT26PC01 | 29 | |
| 13. | Operating Systems Laboratory | BIT26PC02 | 31 | |
| 14. | Automata Theory | BIT26PC03 | 33 | |
| 15. | Seminar | BIT26PC04 | 35 | |
| 16. | DevOps Laboratory | BIT26VS01 | 37 | |
| 17. | Programme Elective Course (PEC) 2 : Deep Learning | BIT26PE01 | 40 | |
| 18. | Programme Elective Course (PEC) 2 Laboratory : Deep Learning Laboratory | BIT26PE04 | 42 | |
| 19. | Programme Elective Course (PEC) 2 : Full Stack Development | BIT26PE02 | 44 | |
| 20. | Programme Elective Course (PEC) 2 Laboratory : Full Stack Development Laboratory | BIT26PE05 | 46 | |

Chairman
BoS, Information Technology
PCET's, Pimpri Chinchwad College of Engineering
Sector No. 26, Pradhikaran, Nigdi, Pune-44

| | | | |
|---|---|---|---|
| 21. | Programme Elective Course (PEC) 2 : Internet of Things | BIT26PE03 | 48 |
| 22. | Programme Elective Course (PEC) 2 Laboratory : Internet of Things Laboratory | BIT26PE06 | 50 |
| 23. | Programme Elective Course (PEC) 3 : Natural Language Processing | BIT26PE07 | 52 |
| 24. | Programme Elective Course (PEC) 3 Laboratory : Natural Language Processing Laboratory | BIT26PE10 | 54 |
| 25. | Programme Elective Course (PEC) 3 : Software Testing and Quality Assurance | BIT26PE08 | 56 |
| 26. | Programme Elective Course (PEC) 3 Laboratory : Software Testing and Quality Assurance Laboratory | BIT26PE11 | 58 |
| 27. | Programme Elective Course (PEC) 3 : Application of Routing Algorithms | BIT26PE09 | 60 |
| 28. | Programme Elective Course (PEC) 3 Laboratory : Application of Routing Algorithms Laboratory | BIT26PE12 | 62 |

Chairman
BoS, Information Technology
PCET's, Pimpri Chinchwad College of Engineering
Sector No. 26, Pradhikaran, Nigdi, Pune-44

**Approved by Academic Council:**

Chairman
Academic Council
PCET's, Pimpri Chinchwad College of Engineering
Sector No. 26, Pradhikaran, Nigdi, Pune-44

Chairman, Academic Council
Pimpri Chinchwad College of Engineering

PCCOE
"Knowledge Brings Freedom"

**INDEX**


| Sr. No. | Content | Page No. |
|---|---|---|
| 1 | Curriculum Framework | i |
| 2 | Curriculum Structure - B.Tech. Semester V | iv |
| 3 | Curriculum Structure – B.Tech. Semester VI | vi |
| 4 | Course Syllabus of Semester – V Courses | 1 |
| 5 | Course Syllabus of Semester – VI Courses | 28 |
| 6 | Vision and Mission of Information Technology Department | 64 |

**CURRICULUM FRAMEWORK**
**(Regulations 2023)**

## LIST OF ABBREVIATIONS

| Sr. No. | Abbreviation | Type of Course |
|---|---|---|
| 1 | BSC | Basic Science Course |
| 2 | ESC | Engineering Science Course |
| 3 | PCC | Programme Core Course |
| 4 | PEC | Programme Elective Course |
| 5 | MDM | Multidisciplinary Minor |
| 6 | OEC | Open Elective Course |
| 7 | VSEC | Vocational and Skill Enhancement Course |
| 8 | AEC | Ability Enhancement Course |
| 9 | EEM | Entrepreneurship/Economics/Management Course |
| 10 | IKS | Indian Knowledge System |
| 11 | VEC | Value Education Course |
| 12 | ELC | Experiential Learning Courses |
| 13 | LLC | Liberal Learning Courses |

## COURSE WISE CREDIT DISTRIBUTION

| Sr. No. | Type of Course | No. of Courses | Total Credits | |
|---|---|---|---|---|
| | | | No. | % |
| 1 | Basic Science Course | **8** | **14** | 8.75 |
| 2 | Engineering Science Course | **5** | **12** | 7.50 |
| 3 | Programme Core Course | **22** | **46** | 28.75 |
| 4 | Programme Elective Course | **11** | **20** | 12.50 |
| 5 | Multidisciplinary Minor | **6** | **14** | 8.75 |
| 6 | Open Elective Course | **3** | **6** | 3.75 |
| 7 | Vocational and Skill Enhancement Course | **4** | **8** | 5.00 |
| 8 | Ability Enhancement Course | **2** | **4** | 2.50 |
| 9 | Entrepreneurship/Economics/Management Course | **2** | **4** | 2.50 |
| 10 | Indian Knowledge System | **1** | **2** | 1.25 |
| 11 | Value Education Course | **2** | **4** | 2.50 |
| 12 | Experiential Learning Courses | **4** | **22** | 13.75 |
| 13 | Liberal Learning Courses | **2** | **4** | 2.50 |
| | **Total** | **72** | **160** | **100.00** |

## SEMESTER-WISE COURSE DISTRIBUTION

| Course Distribution: Semester Wise | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Sr. No. | Type of Course | No. of Courses / Semester | | | | | | | | Total |
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 1. | Basic Science Course | 4 | 4 | | | | | | | 8 |
| 2. | Engineering Science Course | 3 | 2 | | | | | | | 5 |
| 3. | Programme Core Course | | 1 | 4 | 4 | 5 | 4 | 4 | | 22 |
| 4. | Programme Elective Course | | | | | 2 | 4 | 4 | 1 | 11 |
| 5. | Multidisciplinary Minor | | | 1 | 1 | 2 | 1 | 1 | | 6 |
| 6. | Open Elective Course | | | 2 | 1 | | | | | 3 |
| 7. | Vocational and Skill Enhancement Course | 1 | 1 | 1 | | | 1 | | | 4 |
| 8. | Ability Enhancement Course | | 1 | | 1 | | | | | 2 |
| 9. | Entrepreneurship/Economics/Management Course | | | 1 | 1 | | | | | 2 |
| 10. | Indian Knowledge System | 1 | | | | | | | | 1 |
| 11. | Value Education Course | | | 1 | 1 | | | | | 2 |
| 12. | Experiential Learning Courses | | | | 1 | | | 1 | 2 | 4 |
| 13. | Liberal Learning Courses | 1 | 1 | | | | | | | 2 |
| **Total** | | **10** | **10** | **10** | **10** | **9** | **10** | **10** | **3** | **72** |

## SEMESTER-WISE CREDIT DISTRIBUTION

| Credit Distribution: Semester Wise | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Sr. No. | Type of Course | No. of Credits / Semester | | | | | | | | Total |
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 1. | Basic Science Course | 7 | 7 | | | | | | | 14 |
| 2. | Engineering Science Course | 7 | 5 | | | | | | | 12 |
| 3. | Programme Core Course | | 2 | 8 | 8 | 12 | 8 | 8 | | 46 |
| 4. | Programme Elective Course | | | | | 4 | 8 | 6 | 2 | 20 |
| 5. | Multidisciplinary Minor | | | 2 | 2 | 4 | 2 | 4 | | 14 |
| 6. | Open Elective Course | | | 4 | 2 | | | | | 6 |
| 7. | Vocational and Skill Enhancement Course | 2 | 2 | 2 | | | 2 | | | 8 |
| 8. | Ability Enhancement Course | | 2 | | 2 | | | | | 4 |
| 9. | Entrepreneurship/Economics/Management Course | | | 2 | 2 | | | | | 4 |
| 10. | Indian Knowledge System | 2 | | | | | | | | 2 |
| 11. | Value Education Course | | | 2 | 2 | | | | | 4 |
| 12. | Experiential Learning Courses | | | | 2 | | | 4 | 16 | 22 |
| 13. | Liberal Learning Courses | 2 | 2 | | | | | | | 4 |
| **Total** | | **20** | **20** | **20** | **20** | **20** | **20** | **22** | **18** | **160** |

# Curriculum Structure

## Third Year B.Tech.

## Information Technology

# CURRICULUM STRUCTURE

## Third Year B.Tech. (Information Technology) Semester – V

| Thrid Year B.Tech Information Technology (Regulations 2023) (With effect from Academic Year 2026-27) | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Semester V** | | | | | | | | | | | | | | | | | |
| **Course Code** | **Course Name** | **Credit Scheme** | | | | **Teaching Scheme (Hours/Week)** | | | | | **Evaluation Scheme and Marks** | | | | | | |
| | | | | | | | | | | | **FA** | | **SA** | **TW** | **PR** | **OR** | **Total** |
| | | **L** | **P** | **T** | **Total** | **L** | **P** | **T** | **O** | **Total** | **FA 1** | **FA 2** | | | | | |
| **BIT25PC01** | Artificial Intelligence and Machine Learning | 3 | - | - | **3** | 3 | - | - | 1 | 4 | 20 | 20 | 60 | - | - | - | 100 |
| **BIT25PC06** | Artificial Intelligence and Machine Learning Laboratory | - | 2 | - | **2** | - | 4 | - | - | 4 | - | - | - | 50 | 50 | - | 100 |
| **BIT25PC07** | Design and Analysis of Algorithms | 4 | - | - | **4** | 4 | - | - | 1 | 5 | 20 | 20 | 60 | - | - | - | 100 |
| **BIT25PC04** | Design and Analysis of Algorithms Laboratory | - | 1 | - | **1** | - | 2 | - | - | 2 | - | - | - | 25 | 25 | - | 50 |
| **BIT25PC05** | Software Engineering and Modeling Laboratory | - | 2 | - | **2** | - | 4 | - | - | 4 | - | - | - | 50 | - | 50 | 100 |
| - | Programme Elective Course (PEC) 1 | 3 | - | - | **3** | 3 | - | - | 1 | 4 | 20 | 20 | 60 | - | - | - | 100 |
| - | Programme Elective Course (PEC) 1 Laboratory | - | 1 | - | **1** | - | 2 | - | - | 2 | - | - | - | 50 | - | - | 50 |
| - | Multidisciplinary Minor (MDM) III | 3 | - | - | **3** | 3 | - | - | - | 3 | 20 | 20 | 60 | - | - | - | 100 |
| - | Multidisciplinary Minor (MDM) IV | - | 1 | - | **1** | - | 2 | - | - | 2 | - | - | - | 50 | - | - | 50 |
| **Total** | | **13** | **7** | **0** | **20** | **13** | **14** | **0** | **3** | **30** | **80** | **80** | **240** | **225** | **75** | **50** | **750** |

*Multidisciplinary Minor-III and IV: kindly refer to the booklet of Multidisciplinary Minor

**Note:** Students are required to ensure that the same course is not selected under multiple categories such as Core Courses, Programme Electives, Open Electives, or through the Minor Degree in Multidisciplinary Studies. Each course should be credited only once towards the degree requirements

**L**-Lecture, **P**-Practical, **T**-Tutorial, **FA**–Formative Assessment, **FA1**-Formative Assessment 1, **FA2**-Formative Assessment 2, **SA**-Summative Assessment, **TW**-Term Work, **PR**-Practical Exam, **OR**-Oral Exam. **O**-Other i.e. self directed learning, self study, outside the class efforts.

## List of Programme Elective Courses 1

| Course Code | Course Name | |
|---|---|---|
| **BIT25PE01** | Fundamentals of Digital Image Processing | Choose any one |
| **BIT25PE02** | Software Project Management | |
| **BIT25PE03** | Engineering a Cloud | |

## List of Programme Elective Courses 1 Laboratory

| Course Code | Course Name | |
|---|---|---|
| **BIT25PE04** | Fundamentals of Digital Image Processing Laboratory | Choose any one |
| **BIT25PE05** | Software Project Management Laboratory | |
| **BIT25PE06** | Engineering a Cloud Laboratory | |

PCCOE
"Knowledge Brings Freedom"

# CURRICULUM STRUCTURE

## Third Year B.Tech. (Information Technology) Semester – VI

| Third Year B.Tech Information Technology (Regulations 2023)<br>(With effect from Academic Year 2026-27) | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Semester VI** | | | | | | | | | | | | | | | | | |
| **Course Code** | **Course Name** | **Credit Scheme** | | | | **Teaching Scheme (Hours/Week)** | | | | | **Evaluation Scheme and Marks** | | | | | | |
| | | | | | | | | | | | **FA** | | **SA** | **TW** | **PR** | **OR** | **Total** |
| | | **L** | **P** | **T** | **Total** | **L** | **P** | **T** | **O** | **Total** | **FA1** | **FA2** | | | | | |
| **BIT26PC01** | Operating Systems | 3 | - | - | **3** | 3 | - | - | 1 | 4 | 20 | 20 | 60 | - | - | - | 100 |
| **BIT26PC02** | Operating Systems Laboratory | - | 1 | - | **1** | - | 2 | - | - | 2 | - | - | - | 25 | 25 | - | 50 |
| **BIT26PC03** | Automata Theory | 2 | - | 1 | **3** | 2 | - | 1 | 1 | 4 | 20 | 20 | 60 | - | - | - | 100 |
| **BIT26PC04** | Seminar | - | 1 | - | **1** | - | 2 | - | | 2 | - | - | - | 50 | - | - | 50 |
| - | Programme Elective Course (PEC) 2 | 3 | - | - | **3** | 3 | - | - | 1 | 4 | 20 | 20 | 60 | - | - | - | 100 |
| - | Programme Elective Course (PEC) 2 Laboratory | - | 1 | - | **1** | - | 2 | - | - | 2 | - | - | - | 50 | - | - | 50 |
| - | Programme Elective Course (PEC) 3 | 3 | - | - | **3** | 3 | - | - | 1 | 4 | 20 | 20 | 60 | - | - | - | 100 |
| - | Programme Elective Course (PEC) 3 Laboratory | - | 1 | - | **1** | - | 2 | - | - | 2 | - | - | - | 50 | - | - | 50 |
| - | Multidisciplinary Minor (MDM) V | 2 | - | - | **2** | 2 | - | - | - | 2 | 10 | 10 | 30 | - | - | - | 50 |
| **BIT26VS01** | DevOps Laboratory | - | 2 | - | **2** | - | 4 | - | - | 4 | - | - | - | 50 | - | 50 | 100 |
| **Total** | | **13** | **6** | **1** | **20** | **13** | **12** | **1** | **4** | **30** | **90** | **90** | **270** | **225** | **25** | **50** | **750** |

*Multidisciplinary Minor-V: kindly refer to the booklet of Multidisciplinary Minor

**Note:** Students are required to ensure that the same course is not selected under multiple categories such as Core Courses, Programme Electives, Open Electives, or through the Minor Degree in Multidisciplinary Studies. Each course should be credited only once towards the degree requirements

**L**-Lecture, **P**-Practical, **T**-Tutorial, **FA**–Formative Assessment, **FA1**-Formative Assessment 1, **FA2**–Formative Assessment 2, **SA**-Summative Assessment, **TW**-Term Work, **PR**-Practical Exam, **OR**-Oral Exam, **O**-Other i.e. self directed learning, self study, outside the class efforts.

**Note- "Exit Policy: Available as a separate document"**

## List of Programme Elective Courses 2

| Course Code | Course Name | |
|---|---|---|
| **BIT26PE01** | Deep Learning | Choose any one |
| **BIT26PE02** | Full Stack Development | |
| **BIT26PE03** | Internet of Things | |

## List of Programme Elective Courses 2 Laboratory

| Course Code | Course Name | |
|---|---|---|
| **BIT26PE04** | Deep Learning Laboratory | Choose any one |
| **BIT26PE05** | Full Stack Development Laboratory | |
| **BIT26PE06** | Internet of Things Laboratory | |

## List of Programme Elective Courses 3

| Course Code | Course Name | |
|---|---|---|
| **BIT26PE07** | Natural Language Processing | Choose any one |
| **BIT26PE08** | Software Testing and Quality Assurance | |
| **BIT26PE09** | Application of Routing Algorithms | |

## List of Programme Elective Courses 3 Laboratory

| Course Code | Course Name | |
|---|---|---|
| **BIT26PE10** | Natural Language Processing Laboratory | Choose any one |
| **BIT26PE11** | Software Testing and Quality Assurance Laboratory | |
| **BIT26PE12** | Application of Routing Algorithms Laboratory | |

# *Course Syllabus*

# *Semester V*

| Program: | B.Tech. I.T. | | | | | Semester: | V | |
|---|---|---|---|---|---|---|---|---|
| Course: | Artificial Intelligence and Machine Learning | | | | | Code: | BIT25PC01 | |
| Credits | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Basics of Data Structure and Algorithms
2. Basics of Linear Algebra

**is essential**

**Course Objectives:**

This course aims to enable students

1. To understand the fundamentals of Artificial Intelligence and Machine Learning, and various algorithms.
2. To learn ML strategies for building models from data and evaluating their performance
3. To apply ML algorithms to given data and interpret the outcomes.
4. To design appropriate AI & ML solutions for real-world problems.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Interpret fundamental principles of artificial intelligence.
2. Apply constraint satisfaction and game theory algorithms.
3. Make use of Machine learning concepts and data pre-processing techniques.
4. Use regression techniques and evaluate their performance.
5. Analyze different types of classification models.
6. Apply appropriate clustering techniques and association rule mining.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1 | **Introduction to Artificial Intelligence**<br>Introduction of AI, Intelligent Agents: Agents and Environments, Types of Environment, Structure of Agents; Problem Solving by Search: State Space Search, Uninformed Search Strategies, Problem-Solving Agents, Informed strategies: A* search, heuristic functions, hill-climbing. | 7 |
| 2 | **Constraint Satisfaction and Game Theory**<br>Constraint Satisfaction Problems, Game Theory, Optimal Decisions in Games, Adversarial search- Minimax algorithm, Alpha-beta pruning. | 7 |
| 3 | **Introduction to Machine Learning and Data Pre-processing**<br>Introduction to Machine Learning, Types of Machine Learning, Designing a learning system, Issues in machine learning. Bias, Variance, Underfitting, Overfitting, Training, Testing, Data pre-processing techniques, dimensionality reduction-PCA. | 7 |
| 4 | **Regression**<br>Linear regression with one variable, Cost function, Target Function, Gradient Descent, Gradient Descent for Linear Regression, Linear Regression with Multiple Variables, Gradient Descent for Multiple Variables, Logistic regression. | 7 |

| | | |
|---|---|---|
| 5 | **Classification**<br>Needs and applications of classification, Naïve Bayes Algorithm, K-Nearest Neighbours, Support Vector Machines, Decision Tree, Introduction to ANN, Ensemble Learning, Evaluating classification models performance using Confusion matrix, (Sensitivity, Specificity, Precision, Recall, ROC Curves etc.), Enhancing Performance of classification: Cross-Validation, Sub-Sampling, HyperParameter Tuning Techniques. | 9 |
| 6 | **Clustering and Association Rule Mining**<br>Clustering and Association Rule Mining: Need and applications of clustering, Partitioned methods, Hierarchical methods, Density-based methods. Need and application of Association Rules learning, Basic concepts, Apriori Algorithm, FP-Growth, Performance Measures – Support and Confidence. | 8 |
| | **Total** | **45** |

**Text Books:**

1. Russell, S. and Norvig, P. Artificial Intelligence - A Modern Approach, 4th edition, Prentice Hall
2. E. Alpaydin, “Introduction to Machine Learning”, 4th edition, The MIT Press
3. Mitchell M., T., Machine Learning, McGraw-Hill, 2nd Edition.

**Reference Books:**

1. Andreas Muller and Sarah Guido,” Introduction to Machine Learning with Python: A Guide for Data Scientists”, Shroff/O’Reilly.
2. Aurolien Geron,” Hands-On Machine Learning with Scikit-Learn and TensorFlow, Shroff/O’Reilly.
3. Bishop M., C., Pattern Recognition and Machine Learning, Springer-Verlag (2011) 2nd Edition

**E-Books:**

1. Russell, S. and Norvig, P. Artificial Intelligence - A Modern Approach, 4th edition, Prentice Hall
2. E. Alpaydin, “Introduction to Machine Learning”, PHI, 2nd edition

**E-Resources:**

1. NPTEL Artificial Intelligence : Search Methods For Problem solving https://onlinecourses.nptel.ac.in/noc22_cs67/preview
2. NPTEL Introduction to Machine Learning: https://onlinecourses.nptel.ac.in/noc24_cs51/preview
3. NPTEL Machine Learning, ML: https://onlinecourses.nptel.ac.in/noc24_cs60/preview
4. Coursera: Machine Learning Specialization by Andrew Ng

| **Program:** | B. Tech. I.T. | | | | **Semester:** | V |
|---|---|---|---|---|---|---|
| **Course:** | **Artificial Intelligence and Machine Learning Laboratory** | | | | **Code:** | BIT25PC06 |
| **Credits** | **Teaching Scheme (Hrs./Week)** | | | **Evaluation Scheme and Marks** | | |
| | **Lecture** | **Practical** | **Tutorial** | **TW** | **PR** | **OR** | **Total** |
| 2 | - | 4 | - | 50 | 50 | - | 100 |

**Prior knowledge of:**

1. Basics of Data Structure and Algorithms
2. Basics of Linear Algebra

**is essential**

**Course Objectives:**

This course will enable students

1. Make use of datasets in implementing the machine learning algorithms.
2. Implement the machine learning concepts and algorithms in any suitable language of choice.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Apply AI algorithms in problem-solving.
2. Develop a model using classification, regression and clustering algorithms.
3. Use different AI and ML algorithms to build real-life problems.

**Detailed Syllabus**

| **Assignment No.** | **Suggested List of Assignments (Any Six)** |
|---|---|
| 1 | Implement the Informed Search algorithm for real-life problems. |
| 2 | Write a program to implement a Constraint Satisfaction Problem (CSP) by choosing one application such as N-Queens, Sudoku Solver, Exam Timetable Scheduling, or Hostel Room Allocation |
| 3 | Implementation of games (any one): 8 puzzle, Tic-Tac-Toe, Tower of Hanoi, Water jug problems. |
| 4 | Uber Ride Price Prediction using PCA and EDA:<br>• Perform Exploratory Data Analysis (EDA) on Uber ride data<br>• Use Principal Component Analysis (PCA) to reduce dimensionality<br>• Compare the performance of models with and without PCA<br>• Evaluate models using metrics like $R^2$, RMSE, MAE<br>Suggested dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset |
| 5 | Build a linear regression model for a real-life scenario such as Housing Price Prediction, Student Performance Prediction, Salary Estimation, Sales Forecasting for a Retail Store, or any other domain-specific application. |
| 6 | Develop a Decision Tree model for a real-world classification problem (e.g., churn prediction, loan approval, or disease diagnosis) and interpret the results. |
| 7. | Implement the Bagging technique using the Random Forest algorithm to classify a dataset (e.g., Iris or Titanic) and compare its accuracy with a single Decision Tree model. |
| 8. | Implement the K-Nearest Neighbor (KNN) algorithm to classify a given dataset based on feature similarity. Use distance calculation to find the *K* nearest data points and predict the class by majority voting. For ex. Iris Flower Classification, Student Performance Prediction |

| | |
|---|---|
| 9. | Apply the Naïve Bayes algorithm to a real-world classification problem such as email spam detection, sentiment analysis, or disease diagnosis, and evaluate model performance. |
| 10. | Implement the K-Means algorithm to group data in an unsupervised setting, such as customer segmentation, student performance grouping, and analyze the clustering output. |
| 11. | Apply the Apriori algorithm to discover frequent itemsets and generate association rules from transactional data, such as market basket analysis in a retail store. |
| 12. | Implement the FP-Growth algorithm to find frequent itemsets and generate association rules from large transactional datasets, revealing relationships such as {Bread, Butter} → {Milk} for market basket analysis. |
| 13. | Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use a Support Vector Machine for classification. Analyze its performance.<br>Suggested dataset link:<br>https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv |
| 14. | Given a bank customer, build a neural network-based classifier that can determine whether they will leave the bank or not in the next 6 months.<br>Suggested dataset link:<br>https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling.<br>Perform the following steps:<br>1. Read the dataset.<br>2. Distinguish the feature and target set and divide the data set into training and test sets.<br>3. Normalize the train and test data.<br>4. Initialize and build the model. Identify the points of improvement and implement them.<br>5. Print the accuracy score and confusion matrix |

**Reference Books:**


1. George F Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Edu, 4th Edition.
2. Russell, S. and Norvig, P. 2015. Artificial Intelligence - A Modern Approach, 3rd edition, Prentice Hall.
3. E. Alpaydin, “Introduction to Machine Learning”, PHI, 2005.
4. AurolienGeron, ”Hands-On Machine Learning with Scikit-Learn and TensorFlow, Shroff/O’Reilly”, 2017


**E-sources:**

1. NPTEL Artificial Intelligence: Search Methods For Problem solving https://onlinecourses.nptel.ac.in/noc22_cs67/preview
2. https://ds1-iiith.vlabs.ac.in/exp/depth-first-search/index.html
3. https://ds1-iiith.vlabs.ac.in/exp/breadth-first-search/index.html

| Program: | B. Tech. I.T. | | | | Semester: | | V | |
|---|---|---|---|---|---|---|---|---|
| Course: | Design and Analysis of Algorithms | | | | Code: | | BIT25PC07 | |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 4 | 4 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Basics of trees and graphs
2. Proof Techniques
3. Basics of mathematics

**is essential**

**Course Objectives:**

1. To know the basics of computational complexity of various algorithms.
2. To select appropriate algorithm design strategies to solve real-world problems.
3. To understand the concept of nondeterministic polynomial algorithms.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Analyze the asymptotic performance of algorithms
2. Solve computational problems by applying suitable paradigms of Divide and Conquer or Greedy methodologies
3. Generate optimal solutions by applying Dynamic Programming strategy
4. Apply Backtracking strategies to solve various problems.
5. Apply branch & Bound technique to solve problems
6. Classify P, NP, NP-complete, NP-Hard problems.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1. | **Analysis of Algorithm**<br>Efficiency- Analysis framework, asymptotic notations – big O, theta and omega. Analysis of sorting techniques. Analysis of Non-recursive and recursive algorithms: Solving Recurrence Equations using Masters theorem and Substitution method. Brute Force method: Introduction to Brute Force method & Exhaustive search, Brute Force solution to 8 queens' problem. | 10 |
| 2. | **Divide And Conquer**<br>Divide & Conquer: General method, Binary Search, Merge sort and Quick sort, Finding Max-Min<br>**Greedy Method**<br>Greedy Method: General method and characteristics, Optimal storage on tapes, Fractional Knapsack problem. Minimum Spanning Tree with Prim's method and Kruskal's method, Single Source Shortest Path: Dijkstra's Algorithm. | 12 |
| 3. | **Dynamic Programming:**<br>Dynamic Programming: General strategy, Principle of optimality, Multistage Graph problem, 0/1 knapsack Problem, Bellman Ford Algorithm, Optimal Binary Search Trees, Travelling Salesman Problem. | 10 |

| | | |
|---|---|---|
| 4. | **Backtracking**<br>Backtracking: General method, Recursive backtracking algorithm, Iterative backtracking method. 8-Queen problem, Sum of subsets, Graph coloring | 10 |
| 5. | **Branch-and-bound**<br>The method, Control abstractions for Least Cost Search, Bounding, FIFO branch and bound, LC branch and bound, 0/1 Knapsack problem – LC branch and bound and FIFO branch and bound solution, Traveling salesperson problem- LC branch and bound | 10 |
| 6. | **Complexity Theory:**<br>The classes P, NP, NP complete & reduciability, NP hard satisfiability problem,<br>NP Completeness Proof: Clique problem, Vertex cover problem.<br>Approximation Algorithms: Vertex Cover, Travelling salesman | 8 |
| | **Total** | **60** |

**Text Books:**

1. E. Horowitz, S. Sahani, S. Anderson-Freed "Fundamentals of Data Structures in C", Universities Press, 2008.
2. Thomas H Cormen and Charles E.L Leiserson, Introduction to Algorithm, PHI, ISBN:81-203-2141-3.

**Reference Books:**

1. Ellis Horwitz, SartajSahni, RajSekaran. Fundamentals of computer algorithms University Press.
2. G. A.V, PAI, “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6.
3. Gilles Brassard, Paul Bratle, Fundamentals of Algorithms, Pearson, ISBN 978-81-317-1244.
4. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.

**E-Resources:**

1. https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/

"Knowledge Brings Freedom"

| Program: | B. Tech. I.T. | | | | Semester: | | | V |
|---|---|---|---|---|---|---|---|---|
| Course: | Design & Analysis of Algorithms Laboratory | | | | Code: | | | BIT25PC04 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 1 | - | 2 | - | - | 25 | 25 | - | 50 |

**Prior knowledge of:**

1. Basics of trees and graphs
2. Proof Techniques
3. Basics of mathematics

**is essential**

**Course Objectives:**

1. To know the basics of computational complexity of various algorithms.
2. To select appropriate algorithm design strategies to solve real-world problems.
3. To understand the concept of nondeterministic polynomial algorithms.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Analyze the asymptotic performance of algorithms
2. Solve computational problems by applying suitable paradigms of Divide and Conquer or Greedy methodologies
3. Generate optimal solutions by applying Dynamic Programming strategy
4. Apply Backtracking strategies to solve various problems.
5. Apply branch & Bound technique to solve problems
6. Classify P, NP, NP-complete, NP-Hard problems.

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments |
|---|---|
| 1 | Design and implement a sorting algorithm using Merge Sort to efficiently arrange customer orders based on their timestamps. The solution should handle a large dataset (up to 1 million orders) with minimal computational overhead. Additionally, analyze the time complexity and compare it with traditional sorting techniques. |
| 2 | **Movie Recommendation System Optimization**<br>A popular OTT platform, StreamFlix, offers personalized recommendations by sorting movies based on user preferences, such as IMDB rating, release year, or watch time popularity. However, during peak hours, sorting large datasets slows down the system.<br>As a backend engineer, you must:<br>• Implement Quicksort to efficiently sort movies based on various user-selected parameters.<br>• Handle large datasets containing of movies while maintaining fast response times |

| | |
|---|---|
| **3** | **Scenario: Emergency Relief Supply Distribution**<br>A devastating flood has hit multiple villages in a remote area, and the government, along with NGOs, is organizing an emergency relief operation. A rescue team has a limited-capacity boat that can carry a maximum weight of W kilograms. The boat must transport critical supplies, including food, medicine, and drinking water, from a relief center to the affected villages.<br>Each type of relief item has:<br>● A weight (wi) in kilograms.<br>● Utility value (vi) indicating its importance (e.g., medicine has higher value than food).<br>● Some items can be divided into smaller portions (e.g., food and water), while others must be taken as a whole (e.g., medical kits).<br>As the logistics manager, you must:<br>1. Implement the Fractional Knapsack algorithm to maximize the total utility value of the supplies transported.<br>2. Prioritize high-value items while considering weight constraints.<br>3. Allow partial selection of divisible items (e.g., carrying a fraction of food packets).<br>4. Ensure that the boat carries the most critical supplies given its weight limit W. |
| **4** | **Scenario: Smart Traffic Management for Emergency Vehicles**<br>A smart city is implementing an intelligent traffic management system to assist ambulances in reaching hospitals as quickly as possible. The city's road network is represented as a graph, where:<br>● Intersections (junctions) are nodes.<br>● Roads between intersections are edges with weights representing travel time (in minutes) considering traffic congestion.<br>An ambulance is currently at Source (S) and needs to reach the nearest hospital (Destination D) in the shortest possible time. Due to dynamic traffic conditions, the weight of each road segment may change in real time.<br>As a transportation engineer, you are assigned to:<br>1. Implement Dijkstra's algorithm to find the shortest path from the ambulance's current location (S) to all possible hospitals.<br>2. Account for dynamic weight updates as traffic conditions change.<br>3. Optimize the system to work efficiently for a large city with thousands of intersections and roads.<br>4. Provide a visual representation of the optimal path for navigation.<br>Expected Outcome:<br>The system should suggest the quickest route for the ambulance, updating dynamically based on real-time traffic conditions, ensuring minimal response time to emergencies. |
| **5** | **Scenario:**<br>A logistics company, SwiftCargo, specializes in delivering packages across multiple cities. To optimize its delivery process, the company divides the transportation network into multiple stages (warehouses, transit hubs, and final delivery points). Each package must follow the most cost-efficient or fastest route from the source to the destination while passing through these predefined stages.<br>As a logistics optimization engineer, you must:<br>1. Model the transportation network as a directed, weighted multistage graph with multiple intermediate stages.<br>2. Implement an efficient algorithm (such as Dynamic Programming or Dijkstra's Algorithm) to find the optimal delivery route.<br>3. Ensure that the algorithm scales for large datasets (handling thousands of cities and routes). |

| | |
|---|---|
| | 4. Analyze and optimize route selection based on real-time constraints, such as traffic conditions, weather delays, or fuel efficiency.<br>Constraints & Considerations:<br>● The network is structured into N stages, and every package must pass through at least one node in each stage.<br>● There may be multiple paths with different costs/times between stages.<br>● The algorithm should be flexible enough to handle real-time changes (e.g., road closures or rerouting requirements).<br>● The system should support batch processing for multiple delivery requests. |
| **6** | **Scenario: Disaster Relief Resource Allocation**<br>A massive earthquake has struck a remote region, and a relief organization is transporting essential supplies to the affected area. The organization has a limited-capacity relief truck that can carry a maximum weight of W kg. They have N different types of essential items, each with a specific weight and an associated utility value (importance in saving lives and meeting urgent needs).<br>Since the truck has limited capacity, you must decide which items to include to maximize the total utility value while ensuring the total weight does not exceed the truck's limit.<br>Your Task as a Logistics Coordinator:<br>1. Model this problem using the 0/1 Knapsack approach, where each item can either be included in the truck (1) or not (0).<br>2. Implement an algorithm to find the optimal set of items that maximizes utility while staying within the weight constraint.<br>3. Analyze the performance of different approaches (e.g., Brute Force, Dynamic Programming, and Greedy Algorithms) for solving this problem efficiently.<br>4. Optimize for real-world constraints, such as perishable items (medicines, food) having priority over less critical supplies.<br>Extend the model to consider multiple trucks or real-time decision-making for dynamic supply chain management. |
| **7** | **Scenario: University Timetable Scheduling**<br>A university is facing challenges in scheduling exam timetables due to overlapping student enrollments in multiple courses. To prevent clashes, the university needs to assign exam slots efficiently, ensuring that no two exams taken by the same student are scheduled at the same time.<br>To solve this, the university decides to model the problem as a Graph Coloring Problem, where:<br>● Each course is represented as a vertex.<br>● An edge exists between two vertices if a student is enrolled in both courses.<br>● Each vertex (course) must be assigned a color (time slot) such that no two adjacent vertices share the same color (no two exams with common students are scheduled in the same slot).<br>As a scheduling system developer, you must:<br>1. Model the problem as a graph and implement a graph coloring algorithm (e.g., Greedy Coloring or Backtracking).<br>2. Minimize the number of colors (exam slots) needed while ensuring conflict-free scheduling.<br>3. Handle large datasets with thousands of courses and students, optimizing performance.<br>4. Compare the efficiency of Greedy Coloring, DSATUR, and Welsh-Powell algorithms for better scheduling.<br>Extend the solution to include room allocation constraints where exams in the same slot should fit within available classrooms. |

| | |
|---|---|
| **8** | **Scenario: University Timetable Scheduling**<br>A university is facing challenges in scheduling exam timetables due to overlapping student enrollments in multiple courses. To prevent clashes, the university needs to assign exam slots efficiently, ensuring that no two exams taken by the same student are scheduled at the same time.<br>To solve this, the university decides to model the problem as a Graph Coloring Problem, where:<br>● Each course is represented as a vertex.<br>● An edge exists between two vertices if a student is enrolled in both courses.<br>● Each vertex (course) must be assigned a color (time slot) such that no two adjacent vertices share the same color (no two exams with common students are scheduled in the same slot).<br>As a scheduling system developer, you must:<br>5. Model the problem as a graph and implement a graph coloring algorithm (e.g., Greedy Coloring or Backtracking).<br>6. Minimize the number of colors (exam slots) needed while ensuring conflict-free scheduling.<br>7. Handle large datasets with thousands of courses and students, optimizing performance.<br>8. Compare the efficiency of Greedy Coloring, DSATUR, and Welsh-Powell algorithms for better scheduling.<br>Extend the solution to include room allocation constraints where exams in the same slot should fit within available classrooms. |
| **9** | **Scenario: Optimizing Delivery Routes for a Logistics Company**<br>A leading logistics company, SwiftShip, is responsible for delivering packages to multiple cities. To minimize fuel costs and delivery time, the company needs to find the shortest possible route that allows a delivery truck to visit each city exactly once and return to the starting point.<br>The company wants an optimized solution that guarantees the least cost route, considering:<br>● Varying distances between cities.<br>● Fuel consumption costs, which depend on road conditions.<br>● Time constraints, as deliveries must be completed within a given period.<br>Since there are N cities, a brute-force approach checking all (N-1)!permutations is infeasible for large N (e.g., 20+ cities). Therefore, you must implement an LC (Least Cost) Branch and Bound algorithm to find the optimal route while reducing unnecessary computations efficiently. |

**Reference Books:**


1. Thomas H Cormen and Charles E.L Leiserson, Introduction to Algorithm, PHI, ISBN:81-203-2141-3.
2. G. A.V, PAI, “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6.
3. Gilles Brassard, Paul Bratle, Fundamentals of Algorithms, Pearson, ISBN 978-81-317-1244.
4. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.


**E-Resources:**

1. https://www.udemy.com/course/datastructurescncpp/?couponCode=LEARNNOWPLANS

| Program: | B. Tech. I.T. | | | | | Semester: | | V |
|---|---|---|---|---|---|---|---|---|
| Course: | Software Engineering and Modeling Laboratory | | | | | Code: | | BIT25PC05 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 2 | - | 4 | - | - | 50 | - | 50 | 100 |

**Prior knowledge of:**

1. Basics of Programming Language
2. Problem Solving

**is essential**

**Course Objectives:**

1. Introduce software engineering principles, SDLC, and requirements engineering.
2. Explore object-oriented modeling techniques to real-world problems through case studies and projects.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Select appropriate SDLC models based on project requirements and development constraints.
2. Analyze software requirements and convert SRS into structured design.
3. Build static and dynamic system behaviour using UML diagrams.
4. Develop architectural solutions using object-oriented principles.

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance. Suggested Modelling Tools**:** StarUML, Lucidchart, Draw.io, Visual Paradigm.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments (Assignment no.1 to 20)(Any - 15) |
|---|---|
| 1. | Study and analyze different SDLC models (Waterfall, Agile, Spiral, V-Model). Choose an appropriate model for a software project and justify your choice.<br>Sample Software Project:<br>● Library Management System(Waterfall)<br>● Social Media Application (Instagram, Facebook Clone)(Agile)<br>● Hospital Management System(Spiral)<br>● Online Banking System(V-Model) |
| 2. | Prepare a Software Requirements Specification (SRS) document according to ISO/IEC/IEEE 29148:2018 for an E-Commerce Portal (e.g., Flipkart/Amazon clone), identify and map software process activities (communication, planning, modelling, construction, deployment) to their corresponding phases. Highlight key deliverables at each stage of the development cycle. Ensure it includes functional requirements, non-functional requirements, constraints, system interfaces, and traceability.<br>Conduct stakeholder analysis and perform requirement elicitation for an e-commerce portal using techniques such as brainstorming, MoSCoW prioritization, and interviews. Document the functional requirements, user expectations, and feature prioritization.(Draw Kanno diagram) |

| | |
|---|---|
| 3. | Develop a high-level architectural design for a real-world application using Layered Architecture, MVC, or Microservices. For a Health Monitoring System, choose and implement a suitable software architecture (e.g., Layered, MVC, or Microservices). Create an architecture diagram and document rationale for your architectural decisions based on scalability and maintainability. |
| 4. | Identify actors and functional use cases for a Hospital Management System. Prepare a detailed Use Case Diagram and complete Use Case Specifications for major scenarios such as patient registration, doctor consultation, and billing using UML 2.0 notations. |
| 5. | Based on the use case model of a Social Media Platform (e.g., Instagram clone), identify domain classes, relationships, and attributes. Create an Analysis Class Diagram using UML 2.0 that captures the conceptual structure of the system. |
| 6. | Evolve the Analysis Model into a Design Class Model for an Online Banking System. Incorporate Object Constraint Language (OCL) expressions to define class invariants and constraints (e.g., "balance ≥ minimum_balance"). Use UML 2.0 for diagramming |
| 7. | i) Develop a modular software application using the Component-Based Development (CBD) approach by integrating reusable software components (e.g., login system, payment gateway, database module).<br>ii) Design a Component-Based Architecture for an E-Wallet App integrating modules like login, wallet transfer, payment gateway, and transaction history. Prepare a Component Diagram that shows reusable modules and interface dependencies. |
| 8. | For an Online Examination System, identify at least three major scenarios such as user login, starting an exam, and submitting answers. Develop detailed Sequence Diagrams for each scenario using UML 2.0 advanced interaction notations. |
| 9. | Model the dynamic behaviour of an ATM System using a State Chart Diagram. Identify possible object states, events, transitions, and guard conditions. Use UML 2.0 advanced behavioral modelling notations. |
| 10. | Analyze an existing software system, identify performance bottlenecks, and apply code refactoring techniques to optimize efficiency and maintainability. Implement profiling tools like New Relic, PyCharm Profiler, or Chrome DevTools to measure performance improvements. |
| 11. | For a University Course Registration System, identify key activity states and action flows. Draw an Activity Diagram with swimlanes to depict responsibilities of the student, registrar, and academic advisor using UML 2.0 notations. |
| 12. | Capture Functional Requirements with Use Cases for the project / problem statement. Identify actors. List out major use cases and draw detailed Use Case Diagrams using UML2.0 notations. |
| 13. | **Build Analysis Model**<br>MediCare Solutions is developing a **Hospital Management System** to streamline patient care and operations. As a software analyst, your task is to create an **Analysis Class Model** using UML 2.0. The model should capture key entities like Patients, Doctors, Appointments, and Billing, detailing their attributes, methods, and relationships. This model will guide the system's design and development, ensuring clear understanding among stakeholders. |
| 14. | **Build Design Model from Analysis Model**<br>Identify Design classes/ Evolve Analysis Model. Use advanced relationships. Draw Design class Model using Object Constraint Language(OCL) and UML2.0 Notations |
| 15. | Identify Activity states and Action states. Draw Activity diagram with Swim lanes using UML2.0 Notations for major Use Cases |
| 16. | **Build Interaction Model – Sequence Model**<br>Identify at least 3 major scenarios (sequence flow) for your system. Draw Sequence Diagram for every scenario by using advanced notations using UML2.0. |
| 17. | **Build Behavioural Model – State Model**<br>Identify States and events for your system. Identify state transitions and Guard conditions. Draw State chart diagram with advanced UML 2 notations. |

| | |
|---|---|
| 18. | **Testing-**<br>i) For a Library Management System implemented in Java or Python, write unit tests using JUnit (Java) or PyTest (Python) for core methods like issueBook(), returnBook(), and calculateFine(). Demonstrate successful execution and code coverage.<br>ii) Conduct functional and non-functional testing of a Social Media App prototype. Focus on functionality, usability, and responsiveness. Document your observations with screenshots and metrics using tools such as Selenium or Postman |
| 19. | Design a deployment pipeline for an Online Appointment Booking System using DevOps principles. Propose a CI/CD strategy that ensures automation, version control, rollback, and scalability of the software deployment. |
| 20. | Analyze the performance of a legacy University ERP System using profiling tools like PyCharm Profiler or Chrome DevTools. Identify bottlenecks, refactor inefficient code, and compare performance metrics before and after optimization. |
| 21. | **Mini Project:**<br>Develop a mini-project by applying software engineering principles and object-oriented modeling concepts, demonstrating the complete software development lifecycle as part of the laboratory work. Also design and implement a deployment strategy for a software prototype that ensures automation, scalability, and reliability. This involves utilizing industry best practices and tools to create a robust deployment pipeline. |

**Text Books:**

1. Craig Larman, *"Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*", 2nd Edition, ISBN: 9780130925695
2. Ian Sommerville, *"Software Engineering"* ; 10th Edition, Pearson Education; ISBN 10: 1-292-09613-6, 2017.
3. Roger S. Pressman, *"Software Engineering: A Practitioner's Approach"* 9/e, ISBN10: 1260922510, 2020

**Reference Books:**

1. Dan Pilone, Neil Pitman ―UML 2.0 in a Nutshell‖, (In a Nutshell (O'Reilly))

**E-Resources:**

1. https://onlinecourses.nptel.ac.in/noc20_cs68
2. https://onlinecourses.nptel.ac.in/noc20_cs84/preview
3. https://archive.nptel.ac.in/noc/courses/noc21/SEM2/noc21-cs57/

# *Programme Electives*

# *Semester V*

| Program | B. Tech. I.T. | | | | | Semester | V | |
|---|---|---|---|---|---|---|---|---|
| Course | PEC-1:Fundamentals of Digital Image Processing | | | | | Code | BIT25PE01 | |
| Credits | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Boolean Algebra

**is essential**

**Course Objectives:**

1. To know about fundamentals of Digital Image Processing
2. To explore Image Processing Applications

**Course Outcomes:**

After learning the course, the students will be able to:

1. Describe the fundamentals of a digital image processing system.
2. Apply spatial domain enhancement techniques for enhancing images
3. Apply frequency domain methods such as Fourier Transform and filtering for enhancing images.
4. Evaluate image restoration techniques for noise reduction and blur removal using appropriate filtering methods.
5. Use the principles of image compression and representation techniques for efficient image storage and retrieval
6. Apply segmentation and morphological processing methods to extract meaningful regions from images

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1. | **Introduction to Digital Image Processing**<br>Fundamentals of digital image processing and its uses, fundamental steps in digital image processing, components of image processing system, elements of visual Perception, Image sensing and Acquisition, Image sampling and quantization,Basic Relationship between Pixels, basic mathematical tools used in Image processing, Color models: RGB, CMY, HSV, YCbCr | 5 |
| 2. | **Image Enhancement in the Spatial Domain**<br>Intensity transformations, Histogram processing:Histogram equalization, Histogram matching (specification), Spatial filtering:Smoothing filters: mean, GaussianSharpening filters: Laplacian, gradient-based (Sobel, Prewitt), High-boost filtering and edge enhancement. | 7 |
| 3. | **Image Enhancement in the Frequency Domain**<br>1D and 2D Fourier Transform and its properties, Discrete Fourier Transform (DFT), Frequency domain filtering: Ideal low-pass and high-pass filters, Butterworth and Gaussian filters, Convolution theorem and correlation, Homomorphic filtering (illumination-reflectance separation) | 8 |
| 4. | **Image Restoration and Denoising**<br>Image degradation model, Noise models: Gaussian, salt & pepper, speckle Restoration techniques: Inverse filtering, Wiener filtering, Constrained least squares filtering, Non-linear denoising: Median filtering, Adaptive filters | 8 |

| | | |
|---|---|---|
| 5. | **Image Compression and Representation**<br>Basics of image compression, Redundancies: coding, inter-pixel, psycho-visual, Compression techniques: Lossless: Run-Length Encoding, Huffman coding, Arithmetic coding, Lossy: JPEG (DCT-based), Vector Quantization.<br>Image representation: Chain codes, Polygonal approximation, Boundary descriptors (curvature, bending energy), Region descriptors (area, centroid, perimeter) | 8 |
| 6. | **Image Segmentation and Morphological Processing**<br>Line and Edge detection: Roberts, Sobel, Prewitt, Canny, Thresholding: Global thresholding, Otsu's method, Adaptive thresholding, Region-based segmentation: Region growing, Region splitting and merging, Feature Extraction and description Techniques: principal components as feature descriptors, whole Image features, key point orientation, key point descriptors.<br>Morphological operations: Binary morphology: erosion, dilation, opening, closing, Morphological gradient, Hit-or-miss transform. Recent Trends in image processing, Integration of AI and Image processing | 9 |
| | **Total** | **45** |

**Text Books:**

1. Rafael C Gonzalez and Richard E Woods, *Digital Image Processing*,3rd Edition, Publisher : Pearson Education
2. Anil K Jain, *Fundamentals of Digital Image Processing*, Prentice Hall

**Reference Books:**

1. Milan Sonka, Vaclav Hlavac, Roger Boyle, *Image Processing, Analysis AND Machine Vision*, 4th Edition, Cengage Learning
2. Pratt W. K. , *Digital Image Processing* , 3rd Edition, John Wiley & Sons, 2007
3. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing using MATLAB*, 4th Edition, Pearson Education

**E-Resources:**

1. NPTEL Course on "Digital Image Processing" IIT Kharagpur, https://onlinecourses.nptel.ac.in/noc19_ee55/preview

"Knowledge Brings Freedom"

| Program | B. Tech. I.T. | | | | Semester: | | | V |
|---|---|---|---|---|---|---|---|---|
| Course | PEC-1 Laboratory: Fundamentals of Digital Image Processing Laboratory | | | | Code: | | | BIT25PE04 |
| Credits | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | OR | PR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Boolean Algebra
2. Programming and Problem Solving

**is essential**

**Course Objectives:**

1. To study the image fundamentals, mathematical transforms necessary for image processing
2. To know sampling and reconstruction procedures

**Course Outcomes:**

After learning the course, the students will be able to:

1. Apply enhancement techniques to images
2. Experiment image filtering and image restoration to improve the image quality
3. Analyze the effect of compression on image datasets

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance. Course Owner can decide any suitable platform for the suggested list of assignments as per the requirement. Instructor may use any programming paradigm and suitable platform for assignment implementation.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments |
|---|---|
| 1. | Read an 8 bit image and then apply different image enhancement techniques: (a) Brightness improvement (b) Brightness reduction (c) Thresholding (d) Negative of an image (e) Log transformation (f) Power Law transformation. |
| 2. | Read an image, plot its histogram then do histogram equalization. Comment about the result. Implement various spatial domain and frequency domain filters. |
| 3. | Perform image sampling and quantization on set of images and analyze the sample values in sampling process and find different quantized levels in color models. |
| 4. | Consider any suitable set of image dataset with enough size of sample space to evaluate the effect of noise models. Implement noise models like Gaussian, Salt & Pepper, and Impulse and generate their histograms. |

| | |
|---|---|
| 5. | Perform Contrast stretching of a low contrast image to enhance the image quality and compare their Histogram. Perform Histogram Equalization to create uniform distribution of pixel intensities. |
| 6. | Consider any image dataset with raw images containing noise, distortion etc. Apply filtering techniques and implement Inverse filter and Wiener filter over the set of images in the dataset for image restoration and comment on these two filtering effects. |
| 7. | Implement Image compression using DCT Transform and Huffman Coding. Perform all these operations on any open source image dataset and analyse the effect of lossy and lossless image compression. Prepare a report on the same and suggest the suitable technique for image compression based on the samples in dataset. |

**Reference Books:**


1. Rafel C. Gonzalez and Richard E. Woods, Digital Image Processing Using MATLAB, Pearson Education
2. M.K. Bhuyan , " Computer Vision and Image Processing: Fundamentals and Applications", CRC Press, USA, ISBN 9780815370840


**E-Resources:**

1. NPTEL Course on "Image Processing using Python", https://onlinecourses.swayam2.ac.in/nou23_cs15/preview
2. NPTEL Course on , "Computer Vision and Image Processing – Fundamentals and Applications" https://onlinecourses.nptel.ac.in/noc23_ee39/preview

| Program: | B. Tech. I.T. | | | | Semester : | V | | |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-1: Software Project Management | | | | Code : | BIT25PE02 | | |
| Credits | Teaching Scheme (Hrs. / Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Basics of Software Engineering

**is essential.**

**Course Objectives**

This course aims at enabling students:

1. To introduce the principles, methods and techniques of software project management.
2. To learn risk management and quality assurance in project execution.
3. To explore modern tools for project management and agile practices.

**Course Outcomes**

After learning the course, the students should be able to:

1. Apply key phases of the project life cycle and activities.
2. Design structured project plan with resource estimation
3. Identify project risks and apply appropriate risk management techniques.
4. Analyze the process of change control and its impact on software project success.
5. Evaluate the impact of stress, health, ethics and professionalism on effective team
6. Apply UI/UX design principles to enhance software usability.

**Detailed Syllabus**

| Unit No. | Description | Duration (Hrs.) |
|---|---|---|
| 1 | **Foundations of Software Project Management**<br>Importance of Project Management, Project Life Cycle, Activities Covered by Software Project Management, Project Stakeholders and Success Factors, Traditional vs. Agile Project Management | 5 |
| 2 | **Project Planning and Estimation**<br>Project Planning: Project initiation, Project Planning and Scope Management, Four Steps in Project Cost Management, Creating the Work Breakdown Structure, Scheduling the Task and Developing the Schedule using Gantt Charts, PERT/ CPM, RFP Risk Identification and Management; Project Estimation: Software Project Estimation, Make/Buy Decision COCOMO Model I & II, EOC , FP Based Estimation, Decomposition Techniques, Cost Estimation Tools and Techniques; Project Management: The Management Spectrum, People, Product, Process, Project, The W5HH Principle, Metrics in the Process and Project Domains, Software Measurement: size & function oriented metrics (FP & LOC), Metrics for Project<br>Case Study: Cost-Benefit Analysis using MS Excel | 10 |

| | | |
|---|---|---|
| **3** | **Risk Management**<br>Risk Identification and Classification, Risk Analysis Techniques, Risk Planning, Risk control, Risk Mitigation, Risk Strategies, Qualitative vs. Quantitative Risk Analysis Risk Management Tools: SpiraPlan, Risk Management Studio, GRC Cloud<br>Case Study: Risk Analysis for Academic Project | **8** |
| **4** | **Project Tracking and Control Mechanisms**<br>Monitoring and Control Processes, Data Collection Methods,<br>Visual Project Management: Kanban Boards, Gantt Charts<br>Change Control and Software Configuration Management (SCM)<br>Software Configuration Management Tools, Managing Contracts, The Stages of Contract Management, Challenges of Contract Management, Benefits of Contract Management, Types of Contracts | **8** |
| **5** | **Team and Staff Management**<br>Understanding Behaviour-Organizational Behaviour- Selecting the Right Person for the Job-Instruction in the Best Methods-Motivation-The Oldham-Hackman Job Characteristics Model, Team Building and Team Structures, Stress-Health and Safety, Ethical and Professional Considerations<br>Case Study: Team Management for Academic Project | **6** |
| **6** | **User Interface (UI) and User Experience (UX)**<br>Introduction to User Interface (UI) and User Experience (UX) Design Principles, UI Design Guidelines, Accessibility, Design Thinking Process, User Research and Empathy Mapping, Information Architecture and Wireframing, Prototyping Tools, Usability Testing and Feedback, Industry Best Practices in UI/UX | **8** |
| | **Total** | **45** |

**Text Books:**


1. Roger S. Pressman, Bruce R. Maxim, "Software Engineering: A Practitioner's Approach," McGraw-Hill Education, ISBN 978–0–07–337597–7.
2. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management," McGraw-Hill Education, ISBN-10-0077122798 , ISBN-13-978-0077122799.


**Reference Books:**


1. Ian Sommerville, "Software Engineering," Pearson Education, ISBN 10: 1-292-09613-6, ISBN 13: 978-1-292-09613-1.
2. Pankaj Jalote, "Software Project Management in Practice," Pearson Education, ISBN-10 : 0201737213, ISBN-13 : 978-0201737219.
3. Kenneth S. Rubin, "Essential Scrum: A Practical Guide to the Most Popular Agile Process," Addison-Wesley, ISBN-13: 978-0-13-704329-3 ISBN-10: 0-13-704329-5.


**E-Resources:**

1. Git Official Documentation (https://git-scm.com/doc)
2. Atlassian Jira Documentation (https://www.atlassian.com/software/jira)
3. SpiraPlan Documentation (https://www.inflectra.com/SpiraPlan/)

| Program: | B. Tech. I.T. | | | | Semester: | | | V |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-1 Laboratory: Software Project Management Laboratory | | | | Code: | | | BIT25PE05 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Basic software development concepts

**is essential.**

**Course Objectives:**

1. To understand project planning, cost estimation, and risk management in project execution.
2. To explore modern tools for project management and agile practices.

**Course Outcomes:**

After learning the course, the students will be able to

1. Create a project plan and cost estimation.
2. Identify the risk in project management.
3. Identify the key components of software configuration management.
4. Build a mini-project that applies key Software Project Management (SPM) concepts

**Guidelines:**

- Students will design and develop the mini-project as laboratory work.
- The instructor may set multiple sets of assignments and distribute among batches or initiate and guide students in ideation phase.
- It is appreciated if the assignments are based on real-world problems/applications.

**Detailed Syllabus**

| Assignment No. | Description |
|---|---|
| 1. | **Identify the problem statement.**<br>Functional and non-functional requirements from a given problem statement. Functional and non-functional requirements are the primary components of a Software Requirements Specification<br>● Project life cycle and feasibility study.<br>Address the problem statement and provide a solution/prototype to solve it. |
| 2. | **Software Requirement Analysis**<br>● Identify the functional and non-functional requirements for the problem statement.<br>● Create a Software Requirement Specification (SRS) document for the identified problem statement. (Refer IEEE format) |
| 3. | **Project Planning**<br>● Creating the work breakdown structure.<br>● Project scheduling and management techniques (Gantt Chart, PERT, and CPM) |
| 4. | **Cost Estimation**<br>● To estimate cost, effort, and duration for a software project. (COCOMO Model I &II, Function Point Estimation) |
| 5. | **Risk Management**<br>● Create an RTM(Requirements Traceability Matrix) for a Project<br>● Identify and Mitigate Risks<br>● Evaluate Risks Using a Risk Matrix |

| | |
|---|---|
| 6. | **Implementation (Coding)**<br>● Develop core functionalities based on projects including interface UI/UX design.<br>● Set up a Git Repository for collaboration.<br>● Implement basic project structure (Frontend, Backend, and Database). |
| 7. | **Software Configuration Management (SCM)**<br>● Identify the key components of SCM<br>● Kanban-based task management |
| 8. | **Team and Staff Management**<br>● Defining Roles and Responsibilities<br>● Resource Allocation and Workload Distribution |
| 9. | Build a mini-project that incorporates Software Project Management (SPM) concepts—including software requirements specification, project planning, cost estimation, risk management, implementation, software configuration management, and team coordination. (Incorporate Assignments from 1 to 8) |

**Reference Books:**


1. Lawrence J. Peters 2024, "*Software Project Management: Methods and Techniques*", *Publisher*: CRC Press, *ISBN*: 978-1-032-43057-7
2. "Software Project Management: With PMI, IEEE-CS, and Agile-SCRUM" by Moh'd A. Radaideh (2023), De Gruyter, ISBN: 978-3111206462.


**E-Resources:**

1. Git Official Documentation (https://git-scm.com/doc)
2. Atlassian Jira Documentation (https://www.atlassian.com/software/jira)
3. SpiraPlan Documentation (https://www.inflectra.com/SpiraPlan/)

| Program | B. Tech. I.T. | | | | | Semester: | | V |
|---|---|---|---|---|---|---|---|---|
| Course | PEC-1: Engineering a Cloud | | | | | Code: | | BIT25PE03 |
| Credits | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Computer Network Technology and its applications
2. Computer Architecture and its applications

**is essential**

**Course Objectives:**

1. To understand the basics covering cloud infrastructure and services.
2. To study the architecture and security platforms in the cloud computing environment.
3. To study various deployment strategies.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Illustrate the benefits of cloud computing and its applications with respect to various service models.
2. Apply virtual machine concepts and establish networking.
3. Implement cloud architecture, networking, and storage solutions to meet compliance needs.
4. Categorize and manage various databases in the cloud.
5. Deploy monitoring and alerts in a cloud computing environment.
6. Explore the emerging trends in the Cloud computing environment.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1. | **Introduction to Cloud Computing**<br>History and evolution of cloud computing, Key cloud providers: AWS, Azure, Google Cloud, Cloud deployment models: Public, Private, Hybrid, Multi-cloud, Cloud service models: IaaS, PaaS, SaaS, DaaS, NaaS. | 7 |
| 2. | **Cloud Infrastructure & Networking**<br>Virtual machines, containers, and serverless computing, Storage solutions: Object storage, block storage, file storage, Cloud networking: VPC, Load Balancers, DNS, CDN, Hands-on: Setting up a cloud instance and networking, Virtual machines vs. containers vs. serverless. | 8 |
| 3. | **Cloud Security & Compliance**<br>Identity & Access Management (IAM),Encryption, firewalls, and security groups, Compliance standards: General Data Protection Regulation (GDPR), Health Insurance Portability Accountability Act (HIPAA), System and Organization Controls 2 (SOC 2), Hands-on: Configuring IAM roles and security policies. | 7 |

| | | |
|---|---|---|
| 4. | **Cloud Storage & Databases**<br>Object storage (S3, Blob Storage) vs. Block Storage, Cloud databases: SQL (RDS, Cloud SQL), NoSQL (DynamoDB, CosmosDB), Data backup and disaster recovery strategies, Configuring and querying a cloud database. | 10 |
| 5. | **Cloud Monitoring & Performance Optimization**<br>Cloud monitoring tools: Cloud Watch, Azure Monitor, Stack driver, Logging and alerting best practices, Performance tuning and cost optimization, Setting up monitoring and alerts in a cloud computing environment. | 7 |
| 6. | **Advanced Topics & Emerging Trends**<br>Kubernetes and container orchestration, AI/ML in the cloud (AWS Sage Maker, Azure AI, Google AI), Edge computing and IoT in the cloud, Future trends in cloud computing. | 6 |
| | **Total** | **45** |

**Text Books:**

1. Cloud Computing: Concepts, Technology & Architecture" Authors: Thomas Erl, Ricardo Puttini, Zaigham Mahmood.
2. Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)" Author: Michael J. Kavis.
3. Cloud Security Handbook: Secure Applications in AWS, Azure, and GCP Author: Eyal Estrin.

**Reference Books:**

1. Networking in the Cloud: Understanding Network Services for AWS, Google Cloud, and Azure, Author: Alasdair Gilchrist.
2. Terraform: Up & Running Author: Yevgeniy Brikman.

**E-Resources:**

**MOOC/Video Lectures available at:**

1. https://onlinecourses.nptel.ac.in/noc25_cs11/preview

| Program | B. Tech. I.T. | | | Semester: | | | V |
|---|---|---|---|---|---|---|---|
| Course | PEC-1 Laboratory: Engineering a Cloud Laboratory | | | Code: | | | BIT25PE06 |
| Credits | Teaching Scheme (Hrs./Week) | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | TW | OR | PR | Total |
| 1 | - | 2 | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Computer Network Technology and its applications
2. Fundamentals of Computer Architecture

**is essential**

**Course Objectives:**

1. To equip students with practical cloud computing skills, including setting up virtual machines, configuring cloud storage, networking, security, automation, and deploying cloud-based applications.
2. To Distinguish on how networking in the cloud works.
3. Deploy a basic cloud-hosted website and establish a security mechanism.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Determine a method to get the access of the cloud console and examine the layout.
2. Deploy and connect to a cloud-based virtual machine.
3. Create cloud storage and apply file management techniques.

**Guidelines:** Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments (Any-5) |
|---|---|
| 1. | **Setting Up a Cloud Account**<br>Create an account on a cloud provider (AWS, Azure, or Google Cloud) and explore the dashboard.<br>● Sign up for a Free Tier account (AWS, Azure, or GCP).<br>● Navigate the cloud provider's dashboard.<br>● Set up billing alerts to avoid unexpected charges.<br>● Explore regions and availability zones. |
| 2. | **Launching a Virtual Machine (VM)**<br>Deploy a simple cloud-based server.<br>● Navigate to EC2 (AWS), Virtual Machines (Azure), or Compute Engine (GCP).<br>● Launch a Linux VM (Ubuntu).<br>● Configure a security group (firewall rules) to allow SSH access.<br>● Connect to the VM using SSH (PuTTY or terminal).<br>● Install basic software (Apache, Nginx, etc.). |

| | |
|---|---|
| 3. | **Cloud Storage & File Uploads**<br>Store and retrieve files in the cloud.<br>● Create an S3 Bucket (AWS), Blob Storage (Azure), or Cloud Storage (GCP).<br>● Upload a file (image, document, etc.).<br>● Make the file publicly accessible (optional).<br>● Set up bucket permissions for security. |
| 4. | **Cloud Networking (VPC & Load Balancing)**<br>● Create a Virtual Private Cloud (VPC).<br>● Set up a public and private subnet.<br>● Deploy two VMs in the VPC.<br>● Set up a load balancer for traffic distribution. |
| 5. | **Deploying a Static Website**<br>● Upload an HTML page to S3 (AWS) or Azure Blob Storage.<br>● Enable static website hosting.<br>● Configure permissions to allow public access.<br>● Access the website via a URL. |
| 6. | **Cloud Security Basics (IAM & Permissions)**<br>● Navigate to IAM (Identity & Access Management).<br>● Create a new user and assign roles & permissions.<br>● Test access with different permissions (Admin, Read-only, etc.).<br>● Set up MFA (Multi-Factor Authentication) for security. |
| 7. | **Automating Infrastructure with Terraform**<br>● Install Terraform on a local machine.<br>● Write a basic Terraform script to launch a VM.<br>● Deploy infrastructure with terraform apply.<br>● Destroy infrastructure with terraform destroy. |
| 8. | **Deploying a Simple Cloud-Based Application**<br>● Launch a web server VM.<br>● Store website assets in cloud storage.<br>● Set up a database (RDS, Cloud SQL, or CosmosDB).<br>● Configure DNS & Load Balancer for high availability. |

**Reference Books:**


1. Cloud Computing: Concepts, Technology & Architecture Authors: Thomas Erl, Zaigham Mahmood, and Ricardo Puttini.
2. Architecting the Cloud: Design Decisions for Cloud Computing Service Models Author: Michael J. Kavis.


**E-Resources:**

1. https://www.youtube.com/watch?v=NfnVflt1Jxw
2. https://aws.amazon.com/it-pro/hands-on/
3. https://www.youtube.com/watch?v=_LIuNAu5Ktc

# *Course Syllabus*

# *Semester VI*

| Program: | B. Tech. I.T. | | | | Semester: | | VI | |
|---|---|---|---|---|---|---|---|---|
| Course: | Operating Systems | | | | Code: | | BIT26PC01 | |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Computer Organization and Architecture.
2. Fundamentals of Data Structures

**is essential**

**Course Objectives:**

1. To learn the basic concept and functions of the operating system.
2. To understand the concept of Process and Thread management including resource management
3. To understand I/O Management and File System.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Explain the role of the Operating System.
2. Demonstrate the concepts of Process / Thread Scheduling.
3. Make use of Process Synchronization Techniques.
4. Make use of the memory management techniques.
5. Compare the various techniques of I/O management, disk scheduling and file system.
6. Discuss issues and solutions in OS Security.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1. | **Introduction**<br>Basic organization of hardware components, Role of OS relative to hardware functionality with examples related to the von Neumann architecture. Operating System Objectives, The systems stack and role of OS, resources, abstractions and interfaces The Evolution of Operating System, OS Design Considerations for Multiprocessor and Multicore OS, Architectures of Operating System: Monolithic, Microkernel, Exokernel, OS Examples of different types of OSes (RTOS, Desktop, mobile etc.), | 6 |
| 2. | **Process Management**<br>Process abstraction- program vs. process, Process Control Block (PCB). Design of system calls - invocation and basic OS handling Process control system calls - fork, wait, exec, getpid, getppid and variants<br>The process lifecycle--- source code to execution The OS mode of execution - limited direct execution recap, interrupts, system calls. The process contexts switch mechanism and PCB state.<br>Process Scheduling: Types of Scheduling, Scheduling Criteria, Scheduling Algorithms: First-Come First- Served, Shortest-Job-First, Priority, Round Robin, Case Study - Linux Scheduling | 8 |

| | | |
|---|---|---|
| 3. | **Process Synchronization**<br>Motivation - application, process and OS use cases. Principles of Concurrency, Critical - Section Problem, API Synchronization primitives - limitations of software solutions, atomic instructions, test-and-set, spinlocks, mutexes, condition variables, semaphores. API Case studies - producer consumer, reader-writers<br>Deadlock: Principles of Deadlock, Deadlock Characterization: Necessary Conditions, Resource - Allocation Graph, Methods for Handling Deadlock: Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery, Dining Philosopher Problem Study: Linux OS, Windows OS | 10 |
| 4. | **Memory Management**<br>Address bus and memory access Memory view of a process - heap, stack, code, data Process memory usage requirements The address space abstraction using virtual memory and related system calls (mmap, munmap, sbrk, mprotect) Address translation mechanisms - static mapping, segmentation, paging Page faults, page sharing, read/write permissions, swapping, process vs. OS memory. Memory bookkeeping and management. Case Study: Linux OS, Windows OS | 8 |
| 5. | **Input / Output and File Management**<br>I/O Management: I/O Devices, Organization of the I/O Function, I/O Buffering, Secondary Storage Management: Disk Structure, Disk Scheduling<br>File Management: Overview-Files and File Systems, File structure. File Organization and Access, File Directories, File Sharing, Case Study: Linux File System, Android File System | 8 |
| 6. | **Protection and Security**<br>Goals of protection, Domain of protection, Access matrix, Implementation of access matrix, Revocation of access rights, Security problems, Authentication, Program threats, System threats, Threat monitoring | 5 |
| | **Total** | **45** |

**Text Books:**


1. William Stallings, *Operating System: Internals and Design Principles*, Prentice Hall, 8th Edition,2014, ISBN-10: 0133805913 • ISBN-13: 9780133805918
2. Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*, 2018 Arpaci-Dusseau Books, LLC ISBN:9781985086593
3. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, *Operating System Concepts*, John Wiley & Sons ,Inc., 9th Edition,2012, ISBN 978-1-118-06333-0


**Reference Books:**


1. Tom Adelstein and Bill Lubanovic, *Linux System Administration*, O'Reilly Media, ISBN-10: 0596009526, ISBN-13: 978-0596009526.
2. Arnold Robbins, Nelson H. F. Beebe, *Classic Shell Scripting*, O'Reilly Media, Inc., 2005, ISBN 9780596005955
3. Harvey M. Deitel, *Operating Systems*, Prentice Hall, ISBN-10: 0131828274, ISBN-13: 978-0131828278.
4. Thomas W. Doeppner, *Operating System in depth: Design and Programming*, WILEY, ISBN: 978- 0-471-68723-8.
5. Mendel Cooper, *Advanced Shell Scripting*, Linux Documentation Project.
6. Andrew S. Tanenbaum& Herbert Bos, *Modern Operating System*, Pearson, ISBN-13: 9780133592221, 4th Edition.


**E-Resources:**

1. https://pages.cs.wisc.edu/~remzi/OSTEP/

| Program: | B. Tech. I.T. | | | | Semester: | | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | Operating Systems Laboratory | | | | Code: | | | BIT26PC02 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 1 | - | 2 | - | - | 25 | 25 | - | 50 |

**Prior knowledge of:**

1. C Programming
2. Fundamentals of Data Structures

**is essential**

**Course Objectives:**

1. To introduce and learn Linux commands required for administration
2. To demonstrate the functioning of OS basic building blocks like processes, threads under the LINUX.
3. To demonstrate the functioning of OS concepts in user space like concurrency control (process synchronization, mutual exclusion), CPU Scheduling, Memory Management and Disk Scheduling in LINUX.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Use Linux shell commands and library functions.
2. Implement Resource Management algorithms.
3. Develop solutions for classical synchronization problems using inter-process communication mechanisms.

**Guidelines:**

Faculty should assign different OS assignments to each batch, ensuring varied focus areas to promote deeper understanding and reduce redundancy. Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments (Any 6) |
|---|---|
| 1 | **Application of Basic Linux Commands and Bash Scripting**<br>echo, ls, read, cat, touch, test, loops, arithmetic comparison, conditional loops, grep, sed, find, diff, tac and Bash scripting. |
| 2 | **Process Creation, Demonstration of Zombie and Orphan States**<br>Implement a C program in which the main function accepts the integers to be sorted. Main function uses the FORK system call to create a new process called a child process. Parent process sorts the integers using a sorting algorithm and waits for the child process using WAIT system call to sort the integers using any sorting algorithm. Also demonstrate zombie and orphan states. |
| 3 | **Simulate any 2 CPU Scheduling Algorithms (1 preemptive and 1 non- preemptive) with different arrival times.**<br>A system receives processes at different times. Simulate and compare the performance of:<br>• Round Robin (Preemptive)<br>• Shortest Job Remaining First (Preemptive)<br>• Priority (Preemptive)<br>• Shortest Job First (Non-Preemptive) |

| | |
|---|---|
| | • First Come First Serve(Non-Preemptive)<br>• Priority (Non-Preemptive) |
| 4 | **Implement synchronization using semaphores.**<br>Multiple producer and consumer threads must coordinate access to a shared buffer using **counting semaphores** and **mutex/binary semaphore** to avoid race conditions. |
| 5 | **Manage concurrent access to shared data with writer preference.**<br>Multiple reader and writer threads attempt to access a shared database. Writers must be given priority over readers. |
| 6 | **Implement Banker's Algorithm to avoid deadlocks.**<br>A system has 3 resource types and 5 processes. Each process declares maximum resources and holds some currently. Based on a resource request, determine if the system is in a safe state. |
| 7 | **Solve a classic synchronization problem using semaphores.**<br>Five philosophers are sitting around a table doing one of three things: eating, thinking, or trying to pick up chopsticks. Ensure no deadlocks and prevent starvation. |
| 8 | **Simulate any one memory management technique using page replacement.**<br>Simulate any one page replacement with a minimum of 3 page frames and a reference string of at least 15 pages.<br>• Least Recently Used (LRU)<br>• First In First Out (FIFO)<br>• Optimal algorithm |
| 9 | **Implement and analyze any one disk scheduling technique.**<br>Simulate any one disk scheduling with a given request queue (at least 15 requests) and initial disk head position.<br>• FCFS (First Come First Serve),<br>• SSTF (Shortest Seek Time First),<br>• SCAN<br>• C-SCAN<br>• LOOK<br>• C-LOOK |
| 10 | **Implement Full duplex communication between two independent processes.**<br>First process accepts sentences and writes on first pipe to be read by second process. Second process counts number of characters, number of words and number of lines in accepted sentences, and writes the contents on second pipe to be read by first process and displays on standard output. |
| 11 | **Study Assignment:**<br>a. Demonstrate Linux Compilation process<br>b. Implement a new system call in the kernel space, add this new system call in the Linux kernel by the compilation of this kernel (any kernel source, any architecture and any Linux kernel distribution) and demonstrate the use of this embedded system call using C program in user space. |

**Reference Books:**


1. Das, Sumitabha, *UNIX Concepts and Applications*, TMH, ISBN-10: 0070635463, ISBN-13: 978-0070635463, 4th Edition.
2. William Stallings, *Operating System: Internals and Design Principles*, Prentice Hall, 8th Edition,2014, ISBN-10: 0133805913 • ISBN-13: 9780133805918
3. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, *Operating System Concepts*, John Wiley & Sons ,Inc., 9th Edition,2012, ISBN 978-1-118-06333-0
4. Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*, 2018 Arpaci-Dusseau Books, LLC ISBN:9781985086593
5. Yashwant Kanetkar, *UNIX Shell Programming*, BPB Publication ISBN 81-7029-753-2.

| Program: | B. Tech. I.T. | | | | Semester: | | VI | |
|---|---|---|---|---|---|---|---|---|
| Course: | Automata Theory | | | | Code: | | BIT26PC03 | |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 2 | - | 1 | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Logic and Set Theory
2. Computer Programming and Problem Solving

**is essential**

**Course Objectives:**

1. To understand the basics of automata theory and its operations.
2. To study various automata for different types of language classes.
3. To understand the fundamentals of decidability.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Draw finite automata for computing problems.
2. Convert regular expressions and Finite Automata into each other for the formal languages.
3. Generate the grammar for formal languages.
4. Construct push down automata for formal languages.
5. Design Turing machine for formal languages.
6. Explain the decidable and undecidable problems, analyze complexity classes.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1 | **Finite Automata**<br>**Basic Concepts**: Symbols, Alphabet, Strings, formal and natural languages, Finite State Machine<br>**Finite Automata (FA)**: Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Concept of State Transition Diagram, Construction of DFA, Conversion of NFA with and without epsilon transitions to DFA, Minimization of FA<br>**FA with output**: Moore and Mealy machines- Definition, Construction, Inter-Conversion | 8 |
| 2 | **Regular Expressions**<br>Definition of Regular Expressions (RE), Construction of regular expression of the given language, construction of language from the RE, Conversion of RE to FA, DFA using direct method, Conversion of FA to RE using Arden's theorem, Pumping lemma for Regular Languages, Closure properties of Regular Languages, Myhill-Nerode theorem | 7 |
| 3 | **Grammar and Languages**<br>**Grammar:** Introduction, Chomsky Hierarchy<br>**Regular Grammar (RG)**: Formal definition of RG, Conversion- Left Linear Grammar to Right Linear Grammar and vice versa, RG to FA and FA to RG<br>**Context Free Grammar (CFG):** Formal definition of CFG, Derivation- Leftmost Derivation, Rightmost Derivation, Derivation Tree, Ambiguous CFG, Simplification of CFG, Normal forms- Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Closure properties of Context Free Language | 10 |

| | | |
|---|---|---|
| 4 | **Pushdown Automata**<br>Introduction and definition of Push Down Automata (PDA), Deterministic PDA and Non-Deterministic PDA, Equivalence of Acceptance by Final State and Empty Stack, Conversion of PDA to CFG and Vice Versa | 7 |
| 5 | **Turing Machines**<br>Formal definition of a Turing machine (TM), Church - Turing Thesis, Design of TM, Variants of TM, Multi-tape TM, Universal TM, Comparisons of all automata. | 7 |
| 6 | **Computational Complexity**<br>Decidable problems, recursive languages and recursively enumerable languages, decidable problems related to regular languages and context free grammar, halting problem of TM, un-decidable problems, post correspondence problem, mapping reducibility.<br>Complexity Classes: P, NP - NP Completeness and NP Hard Problems. | 6 |
| | **Total** | **45** |

**Text Books:**


1. John C. Martin, Introduction to Language and Theory of Computation, TMH, 3rd Edition, ISBN: 978-0070660489.
2. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory Languages and Computation, Addison-Wesley, ISBN 0-201-44124-1.
3. Vivek Kulkarni, Theory of Computation, Oxford University Press, ISBN- 13 : 978-0198084587.


**Reference Books:**


1. Michael Sipser, Introduction to the Theory of Computation, CENGAGE Learning, 3rd Edition ISBN-13:978-81-315-2529-6.
2. K.L.P Mishra, N. Chandrasekaran, Theory of Computer Science : Automata, Languages and Computation, Prentice Hall India, 2nd Edition.


**E-Resources:**

1. https://cglab.ca/~michiel/TheoryOfComputation/TheoryOfComputation.pdf
2. https://www.cs.virginia.edu/~robins/Sipser_2006_Second_Edition_Problems.pdf
3. https://nptel.ac.in/courses/106/104/106104148/
4. https://nptel.ac.in/courses/106/104/106104028/

| Program | B. Tech. I.T. | | | | Semester: | | | VI |
|---|---|---|---|---|---|---|---|---|
| Course | Seminar | | | | Code: | | | BIT26PC04 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | OR | PR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Course Objectives:**

1. To perform focused study of technical and research literature relevant to a specific topic.
2. To study, interpret and summarize literature scientifically.
3. To promote learning experiences that are collaborative, interdisciplinary, and focused on long-term engagement.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Develop comprehensive literature review on identified problems to evaluate and compare alternative solutions by selecting the most feasible one.
2. Demonstrate Collaborative learning, Interpersonal Skills, Meta cognitive skills through different seminar activities.
3. Create a well-structured and high-quality draft of paper aligned with the chosen problem statement.

**Guidelines:**

1. The seminar is an important collaborative academic activity that must be completed in accordance with the credit requirements outlined in the academic regulations.
2. A group of students (3 to 4) is required to identify the problem(s) in the domain of Computer Engineering/Information Technology, ideally aligned with current technological advancements and trends.
3. The seminar coordinator, in consultation with a panel of experts, assigns seminar guides based on the registered team's area of interest and the guide's expertise.
4. Students should identify individual seminar topics based on the project undertaken in consultation with the guide.
5. Seminar topics should be based on projects undertaken. Guide should thoughtfully allocate seminar topics on different techniques to solve the given problem (project statement), comparative analysis of the earlier algorithms used or specific tools used by various researchers
6. The group must review sufficient literature (reference books, journal articles, conference papers, white papers, magazines, web resources etc.) in relevant areas on their topic as decided by the guide.
7. Research articles could be referred from IEEE, ACM, Science direct, Springer, Elsevier, IETE,CSI or from freely available digital libraries like Digital Library of India (dli.ernet.in), National Science Digital Library, JRD Tata Memorial Library, citeseerx.ist.psu.edu, getcited.org, arizona.openrepository.com, Open J-Gate, Research Gate, worldwidescience.org etc.
8. The group shall present the study as an individual seminar presentation lasting 20–25 minutes, utilizing appropriate audio/visual aids, and submit a draft of paper less than 10 pages having less than 10% of plagiarism.
9. Students are expected to use open source tools for writing drafts of a paper, citing the references and plagiarism detection.
10. Seminar Review: The Seminar Coordinator, in collaboration with the Head of the Department, will form a review committee consisting of domain experts and senior faculty members.
11. This committee will be responsible for approving the seminar topic.
12. Presentations or discussions may be conducted based on the contents of the selected topic.
13. The committee will also assess the timely progress of the seminar work
14. Each student is required to attend two review sessions as scheduled in the seminar calendar.

**Guidelines for Assessment:**

A panel comprising faculty members, including the assigned guide, will evaluate the seminar work based on the following criteria: relevance and quality of the selected topic, depth and clarity of content, effectiveness of presentation, consistency and punctuality in progress, timely completion of tasks, ability to respond to questions, quality of the draft of the paper and level of active participation.

**Rubrics**

| Sr. No. | Reviews | Rubrics |
|---|---|---|
| 1 | Review-1 (25 Marks) | Identification and clear definition of the problem. (5M) |
| | | Evaluation of scope, feasibility, and well-defined objectives (5M) |
| | | Quality and depth of the literature review (5M) |
| | | Effectiveness in communication, presentation, and regular participation. (5M) |
| | | Question & Answers – (5M) |
| 2 | Review-2 (25 Marks) | Analysis and explanation of algorithms or proposed methods with strong technical understanding. (10M) |
| | | Quality of a draft paper. (5M) |
| | | Effectiveness in communication, presentation, and regular participation. (5M) |
| | | Question & Answers – (5M) |

Criteria for evaluation of Term work (50 Marks)

i. Relevance of topic- 05 Marks
ii. Depth of literature reviewed- 10 Marks
iii. Analysis and explanation of algorithms or proposed methods with strong technical understanding. (10M)
iv. Draft of Paper (Technical Content) - 10 Marks
v. Presentation Slides – 5M
vi. Communication Skills-5M
vii. Question and Answers -5M

| Program: | B. Tech. I.T. | | | | | Semester: | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | DevOps Laboratory | | | | | Code: | | BIT26VS01 |
| Credits | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 2 | - | 4 | - | - | 50 | - | 50 | 100 |

**Prior knowledge of:**

1. Software Development

**is essential**

**Course Objectives:**

1. To learn the need of DevOps for software development practice.
2. To explore the various DevOps tools for software development.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Make use of version control system to efficiently track changes, manage code versions, and collaborate on software development.
2. Apply Continuous Integration (CI) and Continuous Deployment (CD) to automate code integration, deployment.
3. Utilize containerization and Container orchestration tools to package, deploy, and manage applications.
4. Apply Configuration Management**,** Infrastructure as Code (IaC**)**, and Cloud Services for management of IT infrastructure.

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments |
|---|---|
| 1. | Git Installation & Setup<br>a. Install Git on your system.<br>b. Configure Git with your name and email.<br>c. Check Git version and setup verification. |
| 2. | Demonstrate the ability to use different git commands to working with local repository , remote repository and log operation (add, commit, status, log, show, branch, checkout, merge, clone, pull, reset, revert, rebase) |
| 3. | Study Jenkins Architecture and Plugins with Jenkins Installation and Setup. |
| 4. | Create a Freestyle Jenkins job: Configure a job to print "Hello, Jenkins!" in the console output and view logs. |
| 5. | Integrate Jenkins with a Git repository . Configure Jenkins to pull code from GitHub. |

| | |
|---|---|
| 6. | Build Job using Jenkins that compiles a Java project and configure triggers to run the build on every commit. |
| 7. | Study Declarative and Scripted pipeline in Jenkins.<br>Create a simple declarative Jenkins pipeline using a Jenkinsfile that includes stages for build, test, and deploy. |
| 8. | Docker Integration: Set up a Jenkins job that builds a Docker image from a Dockerfile and pushes it to a Docker registry. |
| 9. | Explore the Docker architecture, Docker commands and Dockerfile and containerize a simple Python Application in Docker. |
| 10. | Explore the concept of Container orchestration using Kubernetes with Kubernetes architecture and the key components of a Helm chart (Chart.yaml, values.yaml, templates/)Helm commands: install, upgrade, uninstall, and template. |
| 11. | Discover kubernetes objects (Pod, Deployment, Service,Namespace, ConfigMap, Secret, Persistent Volume) and kubernetes services (ClusterIP, NodePort, LoadBalancer, ExternalName). |
| 12. | Install Minikube. Create and deploy a simple Nginx pod in a Kubernetes cluster. |
| 13. | Explore the different services of AWS. (EC2, S3, AWS Lambda, RDS, ELB,ECS) and create EC2 instance in AWS. |
| 14. | Discover Infrastructure as a Code using Terraform and write a terraform script to create virtual machine EC2. |
| 15. | Prepare a Case Study on Ansible and demonstrate its application. |

**Reference Books:**


1. Pierluigi Riti " Pro Devops with Google Cloud Platform With Docker, Jenkins, and Kubernetes" Apress publication ISBN:978-1-4842-3897-4, 2018
2. Sanjeev Sharma and Bernie Coyne, "DevOps for Dummies", John Wiley & Sons, Inc., 2nd IBM Limited Edition, ISBN: 978-1-119-04705-6
3. Viktor Farcic, "The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices" Createspace Independent Engineering platform, ISBN: 978-1-5239-1744-0, 2016


**E-Resources:**

1. https://www.edureka.co/blog/devops-tutorial
2. https://www.guru99.com/devops-tutorial.html

# *Programme Electives*

# *Semester VI*

| Program | B. Tech. I.T. | | | | Semester | | VI | |
|---|---|---|---|---|---|---|---|---|
| Course | PEC-2: Deep Learning | | | | Code | | BIT26PE01 | |
| Credits | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Machine Learning
2. Python Programming Language

**is essential**

**CourseObjectives:**

1. To understand the fundamentals of deep neural network architectures, including backpropagation, activation functions, and optimization techniques.
2. To develop deep learning models such as CNNs and RNNs for real-world applications.

**CourseOutcomes:**

After learning the course, the students will be able to:

1. Interpret the neural network fundamentals.
2. Analyze neural network optimization techniques.
3. Apply convolutional neural networks for real-world application.
4. Analyze pretrained neural network architectures.
5. Make use of recurrent neural networks and their variants for real-world applications.
6. Explore advanced deep learning techniques.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1. | **Neural Network Fundamentals:** Biological vs. Artificial Neural Networks, History of Neural Networks and Deep Learning. Perceptron Model and Multilayer Perceptron, Activation Functions: Linear, Sigmoid, Tanh, ReLU, Softmax. Gradient Descent, The Delta Rule and Learning Rates. Forward and Backpropagation Algorithm, Loss Functions, Training an MLP: Chain rule, Backpropagation algorithm. | 7 |
| 2. | **Deep Neural Network and Optimization Techniques:** Deep Feedforward Networks, Weight Initialization Techniques, Vanishing and Exploding Gradient Problems, Batch Normalization, Dropout and Regularization, Optimizers: Hill-descent analogy in 2D Optimizers, Hill descent in 3D and contours. Gradient Descent Variants, Nesterov Accelerated Gradient Momentum, AdaGrad, Adadelta, RMSProp, Adam. | 8 |
| 3. | **Convolution Neural Networks (CNN):** Biological inspiration: Visual Cortex, CNN ArchitectureOverview, CNN Components,Convolution operation: kernels, Padding, Strides. Convolution over RGB images, CNN Tuning, Applications of CNNs. | 8 |
| 4. | **Pretrained Networks:** Transfer Learning,Popular Architectures: LeNet,AlexNet, VGGNet, ResNet,MobileNet, Inception Network. | 7 |

| | | |
|---|---|---|
| 5. | **Recurrent Neural Networks (RNNs):**<br>RNN architecture, Training RNNs, Types of RNNs. Need for Long Short-Term Memory (LSTM)/GRU.LSTM architecture, Variants of LSTM,GRU Cell Structure, Deep RNN, Bidirectional RNN. | 8 |
| 6. | **Advanced Deep Learning and Applications:**<br>Attention Mechanism in Deep Learning, Encoder-Decoder Models, Generative Adversarial Networks(GANs),<br>Case Studies: Computer Vision, Natural Language Processing, Healthcare & Medical Diagnosis, Autonomous Vehicles & Robotics | 7 |
| | **Total** | **45** |

**Text Books:**

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Neural Network Fundamentals & Deep Learning*, MIT Press, 2017
2. Nikhil Buduma. *Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms.* O Reilly Media, 2017
3. Josh Patterson & Adam Gibson.*Deep Learning A PRACTITIONER'S APPROACH.* O Reilly Media.

**Reference Books:**

1. Richard Szeliski. *Computer Vision: Algorithms and Applications.* (August 18, 2010 draft)
2. Michael Nielsen. *Neural Networks and Deep Learning*

**E-Resources:**

1. https://onlinecourses.nptel.ac.in/noc20_cs62/preview
2. https://www.deeplearning.ai/courses/deep-learning-specialization/

| Program | B. Tech. I.T. | | | | Semester: | | | VI |
|---|---|---|---|---|---|---|---|---|
| Course | PEC-2 Laboratory: Deep Learning Laboratory | | | | Code: | | | BIT26PE04 |
| Credits | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | OR | PR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Computer Programming and Problem Solving

**is essential**

**Course Objectives:**

1. To demonstrate the fundamentals of deep learning architectures including neural networks, backpropagation, optimization techniques, and activation functions.
2. To develop deep learning models such as CNNs and RNNs for real-world applications using modern frameworks like TensorFlow and PyTorch.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Analyze the impact of different optimization techniques on deep neural networks.
2. Design Convolutional Neural Networks (CNNs) for Classification.
3. Apply Recurrent Neural Networks (RNNs) for Sequential Data Analysis
4. Evaluate Generative Deep Learning Models.

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments |
|---|---|
| 1. | Implement a multilayer perceptron (MLP) model for prediction, such as house prices.<br>a. Perform Exploratory Data Analysis<br>b. Prepare dataset<br>c. Build MLP model<br>d. Evaluate Model performance<br>e. Predict for test data |
| 2. | Design multilayer neural network and analyze the effect of batch normalization and dropout in neural network classifiers. |
| 3. | Build a Multiclass classifier using the CNN model. Use MNIST or any other suitable dataset. |
| 4. | Apply a pre-trained network<br>a) Use any three pre-trained models<br>b) Fine-tune the hyper-parameters and compare their performance for a suitable application. |
| 5. | Perform object detection using YOLO. |

| | |
|---|---|
| 6. | Perform Sentiment Analysis in network graph using RNN/LSTM |
| 7. | Implement Auto-encoders for any one of the following tasks:<br>a) Data Compression<br>b) Image de-noising<br>c) Dimensionality reduction |
| 8. | Design and implement a Deep Convolutional GAN to generate images. |

**Reference Books:**


1. Goodfellow I., Bengio,Y., and Courville, A., "Deep Learning", MIT Press, 2016.
2. Zaccone, G., Karim, M. R., Menshawy, A. "Deep Learning with TensorFlow: Explore neural networks with Python", Packt Publisher, 2017.
3. Francois Chollet, "Deep learning with Python" – Manning Publications, 2017.
4. Explore neural networks with Python", Packt Publisher, 2017
5. Antonio Gulli, Sujit Pal "Deep Learning with Keras", Packt Publishers, 2017.
6. Dr. Adrian Rosebrock *Deep Learning for Computer Vision with.* PYIMAGESEARCH


**E-Resources:**

1. NPTEL: Computer Science and Engineering – NOC : Deep Learning- Part 1
2. Deep Learning - Course (nptel.ac.in)

| Program: | B. Tech. I.T. | | | | Semester: | | VI | |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-2: Full Stack Development | | | | Code: | | BIT26PE02 | |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Problem Solving Skill.
2. Object Oriented Programming concepts.

**is essential**

**Course Objectives:**

1. To learn full stack software development technologies.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Explain full Stack project structure and Maven for managing dependencies
2. Apply RESTful API principles to build and integrate services in full stack applications.
3. Apply security, testing, and optimization in REST APIs.
4. Use web design techniques to create effective user interfaces.
5. Apply frontend-backend integration using REST APIs
6. Discuss deployment of full-stack applications on the cloud.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1 | **Spring Boot fundamentals**<br>Spring and the MVC (Model-View-Controller) architecture, Spring Boot, REST APIs, and Microservices.<br>Spring Initializer, creating a basic RESTful API in Spring Boot, Understanding POM.xml (Project Object Model), Maven and its application, Dependency Management in Maven, Maven Plugins, Goals, and Build Lifecycle, Running and testing the basic Spring Boot application. | 7 |
| 2 | **RESTful APIs and Database Connectivity**<br>Configuration using application. Properties and understanding Dependency Injection<br>Core Spring Boot Annotations: @Controller, @RestController, @Service, @Repository.<br>Database Connectivity: Introduction to Spring Data JPA and Hibernate, Entity class creation, Structuring a RESTful application using Controller-Service-Repository layers.<br>CRUD operations: @GetMapping, @PostMapping, @PutMapping, @DeleteMapping. JSON request/response handling.<br>Writing SQL-like queries with Spring Data JPA. Introduction to Spring Data MongoDB for integrating MongoDB with Spring Boot, Comparing SQL (JPA/Hibernate) and NoSQL (MongoDB) approaches for data modeling. | 8 |

| | | |
|---|---|---|
| 3 | **Spring Security and Backend Testing**<br>Authentication and Authorization. JWT implementation (JSON Web Tokens), Role-Based Access Control (RBAC), Password Encryption using BCrypt, Securing REST APIs<br>Backend Testing:<br>JUnit test cases for Service and Repository layers, Mockito dependency injection, Testing RESTful APIs using MockMvc. Global exception handling using @ControllerAdvice.<br>SLF4J and Logback, API performance optimization. | 8 |
| 4 | **Frontend designing**<br>TypeScript: Introduction to TypeScript (TS), Variables and Constants, Modules in TS.<br>ReactJS: Introduction to ReactJS, React Components, Inter Components Communication, Components Styling, Routing, Redux- Architecture, Hooks- Basic hooks, useState() hook, useEffect() hook, useContext() hook.<br>Angular Version 10+: Angular CLI, Angular Architecture, Angular Project Structure, Angular Lifecycle, Angular Modules, Angular Components, Angular Data Binding, Directives and Pipes, Angular Services and Dependency Injections (DI), Angular Routers, Angular Forms. | 8 |
| 5 | **Connecting frontend with Spring Boot**<br>Connecting the frontend to the Spring Boot backend through REST APIs. Handling Cross-Origin Resource Sharing (CORS) issues, sending JSON data between frontend and backend, and making API calls using Fetch API for POST, GET, PUT, and DELETE requests. Configuring CORS policies in Spring Boot, handling JSON data structures, and managing API errors with try/catch mechanisms. Running the backend and frontend together, testing full stack CRUD functionality, debugging API calls, and fixing UI issues. | 8 |
| 6 | **Deployment and Final testing**<br>Deploying applications on Netlify or Vercel, configuring environment variables for API endpoints, and integrating frontend with backend in a production environment. Final testing and performance optimization for deployment | 6 |
| | **Total** | **45** |

"Knowledge Brings Freedom"

**Text Books:**


1. Juha Hinkula, Full Stack Development with Spring Boot and React. Pakt Publishing, Third Edition April 2022, ISBN: 978-1801816786.
2. Mark Heckler. Spring Boot: Up & Running.(First Edition) O'Reilly Media, February 2021, ISBN: 978-1492076988.
3. Craig Walls. Spring in Action (6th Edition). Manning Publications, April 2022, ISBN: 978-1617297571.
4. Nabendu Biswas. React Projects. BPB Publications, 2022, ISBN: 978-1484278116


**Reference Books:**


1. Sharma Pankaj. *Spring Boot and Microservices.* BPB Publications, 2022, **ISBN:** 978-93-5551-526-0.
2. Ralph Moseley. *Developing Java Applications with Spring and Spring Boot.* Wiley India, 2021, **ISBN:** 978-93-5479-279-2.


**E-Resources:**

1. https://spring.io/guides/gs/spring-boot
2. https://react.dev/learn
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013842656893362176207 88_shared/overview

| Program: | B. Tech. I.T. | | | | Semester: | | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-2 Laboratory: Full Stack Development Laboratory | | | | Code: | | | BIT26PE05 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Problem Solving skill
2. Object Oriented Programming concepts

**is essential**

**Course Objectives:**

1. Develop RESTful web services with testing using Spring Boot.
2. Integrate frontend and database with Java-based backend services.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Set up Spring Boot and build a basic MVC application.
2. Develop database-driven applications with CRUD functionality and database integration.
3. Build and test RESTful APIs.
4. Create a full-stack application with frontend-backend integration.

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance. Suggested tools and technologies: Java, VS Code, IntelliJ, Spring Tool Suite (STS), MySql or Postgresql or NoSQL.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments |
|---|---|
| 1. | Setting Up Development Environment:<br>Install JDK, Maven, IDE (e.g., IntelliJ/Eclipse) and a relational database (SQL,PostgreSQL/NoSQL). Create a basic "Hello World" Spring Boot application. |
| 2. | Creating a Simple Spring Boot Web Application:<br>Develop a basic Spring Boot MVC application. Understand Controller, Service, and Repository layers. |
| 3. | Configure SQL,PostgreSQL/NoSQL with Spring Boot. Perform CRUD operations. |
| 4. | Build Restful API to execute CRUD Operations |
| 5. | Perform at least 4 test cases to validate the API for the assignment 4 using Postman. |
| 6. | Design and connect a simple frontend (React / Angular) with Spring Boot using REST APIs for The case study selected in Assignment 4. |
| 7. | Integrate the frontend developed in earlier assignments (React or Angular) with the Spring Boot-based backend. Complete the assignment using integration testing (Mockito), handling success/failure states/, security and showcasing full-stack functionality. |

**Mini Project**

Design and develop a full stack web application that allows users to perform CRUD operations on a selected domain (e.g., Task Management, Online Bookstore, Student Portal, Inventory Management, or Event Tracker), with user authentication, frontend-backend integration, and deployment on a public platform.

**Recommended Technology Stack**:

- Backend: Java, Spring Boot, Spring Data JPA
- Frontend: ReactJS or Angular
- Database: H2 (in-memory) or MySQL/PostgreSQL/MongoDB for NoSQL use cases
- Deployment: Netlify/Vercel (frontend), local/Render/other (backend)

**Reference Books:**


1. Craig Walls. *Spring in Action (6th Edition).* Manning Publications, April 2022, **ISBN:** 978-1617297571.
2. Mark Heckler. *Spring Boot: Up & Running(First Edition).* O'Reilly Media, February 2021, **ISBN:** 978-1492076988.


**E-Resources:**

1. https://docs.spring.io/spring-framework/reference/index.html
2. https://react.dev/learn
3. https://angular.dev/tutorials

| Program: | B. Tech. I.T. | | | | | Semester: | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-2: Internet of Things | | | | | Code: | | BIT26PE03 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Basic knowledge of C Programming & Python

**is essential**

**Course Objectives:**

1. Understand the fundamentals and security of IoT and embedded systems including essence, design strategy, and process modeling.
2. Interpret the role of sensors, actuators, communication models, and protocols in IoT systems.
3. Implement low-cost embedded IoT systems and integrate Cloud, Web, and Edge technologies for real-time solutions.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Explore the various concepts, terminologies, and architecture of IoT systems.
2. Interpret the role of sensors, actuators, communication models, and protocols in IoT systems.
3. Distinguish horizontal and vertical applications and identify common underpinning technologies.
4. Apply IoT security protocols to establish secure communication channels between IoT devices.
5. Analyze WoT and IoT in terms of architectures, platforms, and cloud standards for scalable IoT solutions.
6. Apply edge computing concepts and technologies in real-time IoT solutions.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1. | **Introduction to Internet of Things and Embedded System**<br>Internet of Things: Vision, Emerging Trends, Economic Significance, Technical Building Blocks, Physical design of IoT, Things of IoT, IoT Protocols, Logical design of IoT, IoT functional blocks, IoT communication models, IoT Communication APIs, IoT enabling technologies, IoT levels and deployment templates, IoT Issues and Challenges, Applications. Embedded Systems: Application Domain and Characteristic of Embedded System, Real time systems and Real time scheduling, Processor basics and System-On-Chip, Introduction to ARM processor and its architecture. IoT: Definition and characteristics of IoT. | 7 |
| 2. | **Sensing and Actuation**<br>Role of Sensors and Actuators in IoT Sensors- Definition, characteristics, types and their working- Soil moisture sensor, DHT11, Ultrasonic sensor, PIR sensor, sound sensor, color sensor, LDR, accelerometer, Gyroscope, magnetometer etc. Actuator- Definition, characteristics, types and their working- LED, Relay, DC motor, LCD etc. | 7 |
| 3. | **Pillars of Embedded IoT and Physical Devices**<br>Horizontal, verticals and four pillars of IoT, M2M: The internet of devices, RFID: The internet of objects, WSN: The internet of transducer, SCADA: The internet of controllers, DCM: Device, Connect and Manage, Device: Things that talk, Connect: Pervasive | 7 |

| | | |
|---|---|---|
| | Network, Mangae: To create business values. IoT Physical Devices and End points: Basic building blocks of and<br>IoT device, Exemplary device: Raspberry Pi, Raspberry Pi interfaces, Programming Raspberry Pi with Python, Other IoT Devices. | |
| 4. | **IoT Protocols and Security**<br>Protocol Standardization for IoT, Efforts, M2M and WSN Protocols, SCADA and RFID Protocols, Issues with IoT Standardization, Unified Data Standards, Protocols – IEEE 802.15.4, BACNet Protocol, Modbus, KNX, Zigbee Architecture, Network layer, APS layer IoT Security: Vulnerabilities of IoT, Security Requirements, Challenges for Secure IoT, Threat Modeling, Key elements of IoT Security: Identity establishment, Access control, Data, and message security ,non-repudiation and availability, Security model for IoT. | 8 |
| 5. | **Cloud and Web of Things**<br>Web of Things versus Internet of Things, Two Pillars of the Web, Architecture Standardization for WoT, Platform Middleware for WoT, Unified Multitier WoT Architecture, WoT Portals, and Business Intelligence. Cloud of Things: Grid/SOA and Cloud Computing, Cloud<br>Middleware, Cloud Standards – Cloud Providers and Systems, Mobile Cloud Computing, The Cloud of Things Architecture. | 8 |
| 6. | **Edge Computing in IoT**<br>Edge Computing vs Cloud and Fog Computing, Role and Need for Edge in IoT, Edge Architecture: Device, Edge, and Cloud Layers, Edge Devices – Raspberry Pi, Jetson Nano, Arduino, Edge OS – Ubuntu Core, Balena OS. Edge AI – Tensor Flow Lite, Edge Impulse, Real-time Analytics. Edge Middleware – AWS Green grass, Azure IoT Edge. Security – Identity, Access, Lightweight Encryption. Applications – Smart Cities, Industrial IoT, Smart Agriculture, Healthcare. | 8 |
| | **Total** | **45** |

**Text Books:**


1. ArshdeepBahga,VijayMadisetti,―InternetofThings–Ahands-onapproach‖, Universities Press, ISBN: 0: 0996025510, 13: 978-0996025515
2. Honbo Zhou, ―The Internet of Things in the Cloud: A Middleware Perspective‖, CRC Press, 2012. ISBN : 9781439892992
3. Dieter Uckelmann, Mark Harrison, Florian Michahelles, ―Architecting the Internet of Things‖, Springer, 2011. ISBN: 978-3-642-19156-5
4. LylaB.Das,―EmbeddedSystems:AnIntegratedApproach‖Pearson,ISBN:9332511675, 9789332511675


**Reference Books:**


1. David Easley and Jon Kleinberg, ―Networks, Crowds, and Markets: Reasoning About a Highly Connected World‖, Cambridge University Press, 2010, ISBN:10: 0521195330
2. Olivier Hersent, Omar Elloumi and David Boswarthick, ―The Internet of Things: Applications to the Smart Grid and Building Automation‖, Wiley, 2012,9781119958345
3. Olivier Hersent, David Boswarthick, Omar Elloumi , ―The Internet of Things – Key applications and Protocols‖, Wiley, 2012, ISBN:978-1-119-99435-0
4. Barrie Sosinsky, ―Cloud Computing Bible‖, Wiley-India, 2010.ISBN : 978-0-470- 90356-8


**E-Resources:**

3. https://www.coursera.org/specializations/iot
4. https://www.coursera.org/learn/iot

| Program: | B. Tech. I.T. | | | | | Semester: | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-2 Laboratory: Internet of Things Laboratory | | | | | Code: | | BIT26PE06 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Basic knowledge of C Programming & Python

**is essential**

**Course Objectives:**

1. To understand functionalities of various single-boards embedded platforms fundamentals.
2. To develop a comprehensive approach towards building a small low cost embedded IoT system.
3. To implement the assignments based on sensory inputs.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Design the minimum system for sensor-based application
2. Apply suitable sensors and processors to optimize device performance for specific applications
3. Develop a full-fledged IoT application for a distributed environment

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments (Any-12) |
|---|---|
| **Group A** | 1. Study of Raspberry-Pi, Beagle board, Arduino, and other microcontrollers,Connectivity and configuration of Raspberry-Pi /Beagle board circuit with basic peripherals, LEDs and understand the process of OS installation on Raspberry-Pi /Beagle board,GPIO and its use in the program.<br>2. Implement the connectivity of Raspberry-Pi / Arduino with a temperature sensor. Write an application to read the environment temperature. If the temperature crosses a threshold value, the application indicates the user using LEDs.<br>3. Use the connectivity of Raspberry-Pi/ Arduino with IR sensor. Write an application to detect obstacles and notify users using LEDs.<br>4. Interface an ultrasonic sensor with Arduino/Raspberry Pi to measure the distance of an object and display the result on a serial monitor or LCD screen. |
| **Group B** | 1. Apply the connectivity of Raspberry Pi/Arduino with the camera. Write an application to capture and store the image.<br>2. Design an application using Raspberry-Pi Arduino to control the operation of the stepper motor.<br>3. Design and develop a simple web-based interface for Raspberry Pi/Arduino that allows remote control of connected LEDs, demonstrating real-time interaction between hardware and a web application.<br>4. Create a small dash board application to be deployed on the cloud using think speak. Different publisher devices can publish their information and interested applications can subscribe. |

| | |
|---|---|
| **Group C** | 1. Develop a Real-time application like a smart home with the following requirements: When a use enters the house the required appliances like fan, and light should be switched ON. Appliances should also get controlled remotely by a suitable web interface. The objective of this application is for the student to construct a complete Smart application in a group. Upload and retrieve sensor data to and from the cloud.<br>2. Develop a Real-time application like a smart home with the following requirements: If anyone comes at the door the camera module automatically captures his images end sit to the email account of the user or sends a notification to the user. The door will open only after the user's approval.Upload and retrieve sensor data to and from the cloud. |
| **Group D** | 1. Write a program to control the movement of a servo motor using Raspberry Pi/Arduino.<br>2. Interface Bluetooth with Arduino/Raspberry and write a program to send sensor data to smartphone using Bluetooth<br>3. Develop a security system using a PIR sensor to detect motion and trigger an alert, and an RFID-based access control system to allow or deny entry.<br>4. Understand the working of MQTT (Message Queuing Telemetry Transport) and establish communication between publisher and subscriber using Raspberry Pi/Arduino. |

**Reference Books:**


1. Nitesh Dhanjani, ―Abusing the Internet of Things‖, O'REILLY, ISBN13:978-93-5313-217
2. Cuno P fister, ―Getting Started with the Internet of Things , O'REILLY, ISBN: 13:978-93- 53023-413-6
3. Massimo Banziand Michael Shiloh,―Getting Started with Arduino‖, MAKER MEDIA, ISBN: 13:978-93-5110-907-5
4. Don Wilcher,―BASIC ArduinoProjects,MAKER MEDIA, ISBN:13:978-93-5110-503-9
5. CefnHoile, Clare Bowman, Sjoerd Dirk Meijer, Brian Corteil, Lauren Orsini, ―Raspberry Pi and AVR Projects, MAKER MEDIA, ISBN: 13:978-93-5110-914-3
6. Wol from Do not, ―A Raspberry Pi Controlled Robot‖, MAKER MEDIA, ISBN: 13:978-93- 5110-913-6 7.
7. KimmoKarvinen and TeroKarvinen, ―Arduino Bots and Gadgets‖, O'REILLY, ISBN: 13:978-93-5023-374-0\
8. Derek Molley, ―Exploring Beaglebone‖, Willey, ISBN: 978-1- 118-935125.


**E-Resources:**

1. https://www.coursera.org/learn/iot
2. https://www.udemy.com/course/iot-internet-of-things-automation-using-raspberry-pi/

| Program: | B. Tech. I.T. | | | | Semester: | | VI | |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-3: Natural Language Processing | | | | Code: | | BIT26PE07 | |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Data Structures,
2. Probability & Statistics,
3. Machine Learning,
4. Python Programming

**is essential**

**Course Objectives:**

1. Understand fundamental NLP concepts and techniques.
2. Build & evaluate NLP models.
3. Explore advanced NLP applications.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Demonstrate NLP fundamentals, text representation, and linguistic structures.
2. Apply text preprocessing techniques to prepare textual data for NLP tasks.
3. Apply syntactic and semantic parsing for language analysis.
4. Analyze NLP model performance using evaluation metrics.
5. Build deep learning models RNNs, LSTMs, and Transformers for NLP applications.
6. Analyze advanced NLP applications.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1 | **Introduction to Natural Language Processing and Language Representation**<br>Introduction of NLP, Scope and Applications, Linguistic Foundations: Syntax, Semantics, Pragmatics, Morphology, Challenges in NLP: Ambiguity, Polysemy, Coreference, Sarcasm, Language Models: Statistical Models: n-grams, Laplace Smoothing, Neural Language Models: CBOW, Skip-gram, Evaluation: Perplexity, Introduction to BLEU Score , Text Representation: Bag of Words, TF-IDF, Word Embeddings: Word2Vec, GloVe, FastText, Brief introduction to contextual embeddings (ELMo, BERT) | 6 |
| 2 | **Text Preprocessing and Linguistic Features**<br>Tokenization (Rule-based, Regex, BPE), Stopword Removal, Stemming vs Lemmatization, POS Tagging and Chunking, Named Entity Recognition (NER), Feature Engineering for NLP: One-Hot Encoding, TF-IDF, Word Embeddings, POS Tags, and Dependency Features.<br>Case Study: Preprocessing for Text Classification | 6 |
| 3 | **Syntactic and Semantic Processing**<br>Syntax Parsing: Constituency Parsing, Dependency Parsing, Semantic Analysis: Word Sense Disambiguation (Lesk, WordNet), Introduction to Semantic Role Labeling.<br>Sentiment Analysis: Lexicon-based, ML-based (VADER, SVM),<br>Topic Modeling: Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), Introduction to Knowledge Graphs. | 7 |

| | | |
|---|---|---|
| 4 | **Machine Learning in NLP**<br>Text Classification using: Logistic Regression, Naïve Bayes, SVM, Named Entity Recognition and POS using Conditional Random Fields, Feature Extraction Pipelines, Evaluation Metrics: Confusion Matrix, Precision, Recall, F1-score, ROC-AUC Cost Spanning Tree, Applications of graph. | 7 |
| 5 | **Deep Learning for NLP**<br>Neural Networks for Text: RNN, LSTM, GRU, Attention Mechanism, Transformers: Architecture and Self-Attention, Pre-trained Models and Fine-Tuning: BERT, RoBERTa, DistilBERT, GPT-2, Hugging Face Transformers Library, Transfer Learning in NLP Tasks: Sentiment, QA, Fine-tune BERT for sentiment classification. | 10 |
| 6 | **Advanced NLP Applications**<br>Applications: Information Retrieval (TF-IDF + BERT Ranking), Text Summarization: Extractive (TextRank), Abstractive (PEGASUS, T5), Chat bots and Dialogue Systems (Rule-based vs Neural), Machine Translation: From Rule-based to Neural (Seq2Seq, Transformers), Question Answering Systems: SQuAD, BERT QA Pipeline, Explainability and Ethical concerns in NLP. | 9 |
| | **Total** | **45** |

**Text Books:**

1. Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing (3rd ed.). Pearson.
2. Manning, C. D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.

**Reference Books:**

1. Tunstall, L., von Werra, L., & Wolf, T. (2022). Natural language processing with transformers: Building language applications with Hugging Face. O'Reilly Media.
2. Vajjala, S., Majumder, B., Gupta, A., & Surana, H. (2020). Practical natural language processing: A comprehensive guide to building real-world NLP systems. O'Reilly Media.
3. Goyal, P., Pandey, S., & Jain, K. (2018), Deep learning for natural language processing. Apress. https://doi.org/10.1007/978-1-4842-3790-4
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

**E-Resources:**

1. https://onlinecourses.nptel.ac.in/noc23_cs45/preview
2. Coursera
    1. Deep Learning Specialization by Andrew Ng (Sequence Models for NLP)
    2. Natural Language Processing by Deeplearning.AI
3. Fast.ai NLP Course
4. Kaggle Notebooks on NLP
5. YouTube Channels:
    1. Stanford NLP Lectures
    2. Sentdex NLP Tutorials
6. GitHub Repositories:
    1. Hugging Face Transformers
    2. TensorFlow and PyTorch NLP Models.

| Program: | B. Tech. I.T. | | | | Semester: | | | VI |
|---|---|---|---|---|---|---|---|---|
| Course : | PEC-3 Laboratory: Natural Language Processing Laboratory | | | | Code : | | | BIT26PE10 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Practical | Tutorial | Lecture | Other | TW | OR | PR | Total |
| 1 | 2 | - | - | - | 50 | - | - | 50 |

**Prior Knowledge of**

1. Machine Learning basics
2. Python Programming Language
3. Probability

**is essential**

**Course Objectives:**

1. **Gain insights into** fundamental techniques for text preprocessing, POS tagging, and syntactic parsing in NLP.
2. **Apply** word embedding and text similarity models to real-world problems.
3. **Analyze** challenges such as context-awareness, ambiguity, and pragmatic factors in building conversational AI

**Course Outcomes:**

After learning the course, the students will be able to:

1. Perform text preprocessing tasks like tokenization, POS tagging, syntactic parsing.
2. Implement and compare word embedding techniques (Word2Vec, GloVe) for semantic analysis.
3. Analyze the challenges in building context-aware chat bots and suggest appropriate NLP models for intent detection.
4. Construct a sample use-case demonstrating NLP applications in healthcare or education with end-to-end system design

**Detailed Syllabus**

**Suggested List of Assignments**

| | |
|---|---|
| 1 | Implement tokenization, stemming, and lemmatization on a small corpus of text. |
| 2 | Develop a Python script to perform part-of-speech tagging on a short text dataset. |
| 3 | Implement an N-gram language model in Python to analyze a text sequence and predict the most probable next word. |
| 4 | Implement and train basic word embedding using a limited text corpus in Python to investigate semantic similarities and relationships among words. |
| 5 | Evaluate the challenges in building a context-aware chatbot. Suggest suitable NLP models for intent detection. |
| 6 | Design a basic text classification solution in Python leveraging Naive Bayes or SVM for sentiment or topic classification on a small dataset. |
| 7 | Use LSTM networks with Tensor Flow or Py Torch in Python to perform sequence labeling on a small text dataset. |
| 8 | Apply logistic regression in Python to perform sentiment analysis on a sample dataset of tweets or reviews |
| 9 | Train an RNN-based language model in Python to generate text from short input sequences. |
| 10 | Design and implement a solution using word embeddings to tackle a practical, real-time problem. |
| 11 | Apply any NLP technique to extract noun phrases from a paragraph and explain the logic. |

**Text Books:**

1. **Jurafsky & Martin – Speech and Language Processing**: Comprehensive coverage of all major NLP topics including POS tagging, embeddings, parsing, QA, and chatbots.
2. **Manning & Schütze – Foundations of Statistical NLP**: Strong foundation in traditional statistical NLP techniques.
3. **Bird, Klein & Loper – NLP with Python (NLTK Book)**: Hands-on NLP guide using Python and NLTK for preprocessing, parsing, and classification.

**Reference Books:**

1. **Goyal et al. – Deep Learning for NLP**: Focused on Word2Vec, GloVe, and deep learning applications in NLP.
2. **Vajjala et al. – Practical NLP**: Real-world applications including text classification and domain-specific NLP like healthcare.
3. **Sumit Raj – Building Chatbots with Python**: Step-by-step guide for rule-based and ML-based chatbot creation.

**E-Resources:**

1. **Stanford CS224n**: Advanced deep learning NLP course covering embeddings, transformers, and QA systems.
2. **Coursera NLP Specialization (DeepLearning.AI)**: Covers full NLP pipeline from basics to applications like MT and QA.
3. **Fast.ai NLP Course**: Practical transfer learning approaches for NLP using ULMFiT.
4. **SpaCy Documentation**: Industrial-strength NLP toolkit for POS tagging, parsing, and noun phrase extraction

| Program: | B. Tech. I.T. | | | | Semester: | | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-3: Software Testing and Quality Assurance | | | | Code: | | | BIT26PE08 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Software Engineering

**is essential**

**Course Objectives:**

1. To apply the testing strategies and methodologies in projects.
2. To understand test management strategies and tools for testing.
3. To apply software test automation and explore various tools

**Course Outcomes:**

After learning the course, the students will be able to:

1. Illustrate the basics of software testing.
2. Apply black-box and white-box testing techniques to design effective test cases.
3. Select the level of testing and identify the different types of defects and defect tracking concepts.
4. Apply test automation concepts using automation tools
5. Make use of software quality assurance (SQA) tools for effective quality improvement processes in software development
6. Analyze Artificial Intelligence (AI) and Machine Learning (ML) techniques in software testing.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1 | **Software Testing Basics**<br>Introduction to testing as an engineering activity, testing as a process, verification and validation, V-Model of testing, Testing Life Cycle – Roles and activities. Basic definitions – errors, faults, defects, failures, test case, test, test bed, test oracle, test harness, software quality, software quality assurance group. Testing Principles. Testers role in a software development organization. Reviews as a testing activity- Types of reviews-inspections and walk-through, need for review policies, components of review plans. Review Metrics. Introduction to test driven development | 7 |
| 2 | **Testing Techniques**<br>Structural testing, Black box approach: random testing, equivalence class partitioning and boundary value analysis, Cause-effect graphing, state transition testing White box approach: test adequacy criteria, code coverage and control flow graphs, paths, Data flow and loop testing. Mutation testing. Writing Junit tests. | 8 |
| 3 | **Levels of Testing and Defect Management**<br>Levels of Testing- unit testing, integration testing, system testing, performance testing, recovery testing, regression testing, alpha, beta and acceptance testing. Test Planning, Test Reports, and Monitoring Test Effectiveness, Origins of defects, Defect Types, Defect repository and test design, Defect severity, life cycle of defect. Defect Reports- Track, Retest and Close. Defect Injection and prevention | 8 |

| | | |
|---|---|---|
| 4 | **Software Test Automation**<br>Software Test Automation, Skills needed for Automation, Scope of Automation, Design and Architecture for Automation, Requirements for a Test Tool, Challenges in Automation, Automated test generation- using Random testing, Property-Based Testing for Real-Time Systems | 8 |
| 5 | **Software Quality Assurance**<br>SQA basics, Components of the Software Quality Assurance System, Quality Assurance vs. Quality Control, Software Quality Assurance models (e.g., CMM, CMMI, TMMi),Quality metrics and standards (e.g., ISO 9001), Ishikawa's seven Quality tools, Case Tools and their effect on Software Quality, Six Sigma, Continuous Quality Improvement | 8 |
| 6 | **Emerging Trends in Software Testing: Key Roles of Software Testing** in AI/ML.Testing AI/ML is different from Traditional Software Testing, challenges in testing AI/ML or tools used for AI/ML testing, Real time case studies. | 6 |
| | **Total** | **45** |

**Text Books:**


1. IienBurnstein, "Practical Software Testing", Springer Publication. ISBN-81-81-28-089-x 2005
2. Srinivasan Desikan, Gopalaswamy Ramesh, Twenty Third Impression "Software Testing: Principles and Practices", PEARSON ISBN-978-81-7758-121-8, 2016.
3. Stephen Kan, "Metrics and Models in Software Quality", Addison – Wesley, Second Edition. ISBN: 0201729156


**Reference Books:**


1. William E Perry, "Effective Methods for Software Testing", Third Edition, Wiley Publication ISBN-13: 978-0-7645-9837-1.
2. S.A.Kelkar, "Software quality and Testing", PHI Learing, Pvt, Ltd. 2012. ISBN-13. 9788120346284.
3. Daniel Galin, "Software Quality Assurance: From Theory to Implementation", Pearson Addison Wesley. ISBN 0201- 70945- 7


**E-Resources**

1. https://asq.org/quality-resources/seven-basic-quality-tools

| Program: | B. Tech. I.T. | | | | Semester: | | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-3 Laboratory: Software Testing and Quality Assurance Laboratory | | | | Code: | | | BIT26PE11 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Basics of software development and testing.
2. Knowledge of programming languages ( Python or Java)

**is essential**

**Course Objectives:**

1. To learn the fundamental concepts, objectives, and types of software testing
2. To provide practical knowledge of manual and automated testing methods, including black-box and white-box testing.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Apply black-box and white-box testing techniques to evaluate the functionality and structure of software applications.
2. Use industry-standard tools to perform automated functional and performance testing of software systems.
3. Analyze and report software defects using defect tracking tools and apply quality assurance techniques to improve software processes.

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments |
|---|---|
| 1. | Write a black box functional test cases using manual testing for the given application |
| 2. | Write a test case using white box testing for the given code.<br>1. Calculate Cyclomatic complexity<br>2. Control flow testing<br>3. Data flow testing |
| 3. | Implement white box unit test cases using Junit for the given application |
| 4. | Prepare a Defect Tracking Report / Bug Report using any Defect Tracking Tool |
| 5. | Implement an automated test using Selenium WebDriver and WebDriver Manager in Python/Java for a sample or real website, simulating a given scenario and verifying its success. |
| 6. | Perform Black Box testing using automated testing tools on an application. Testing Points to be covered – data driven wizard |

| | |
|---|---|
| 7. | Perform Black Box testing using automated testing tools on an application. Testing Points to be covered –parameterization, exception handling |
| 8. | Perform performance testing using a tool for the given application |
| 9. | Implement and execute REST API tests using the Karate DSL framework, applying behaviour-driven development (BDD) style to validate HTTP endpoints. |
| 10. | Explore the different Ishikawa's Quality tools. Use them to draw a fishbone diagram and Pareto distribution for the given scenario and data set. |

**Text Books:**


1. IienBurnstein, "Practical Software Testing", Springer Publication. ISBN-81-81-28-089-x 2005
2. Srinivasan Desikan, Gopalaswamy Ramesh, Twenty Third Impression "Software Testing: Principles and Practices", PEARSON ISBN-978-81-7758-121-8, 2016.


**Reference Books:**


1. Frank Appel , ―Testing with JUnit‖, Oreilly Publication, ISBN: 9781782166603


**E-Resources**

1. https://asq.org/quality-resources/seven-basic-quality-tools

| Program: | B. Tech. I.T. | | | | Semester: | | VI | |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-3: Application of Routing Algorithms | | | | Code: | | BIT26PE09 | |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | FA | | SA | Total |
| | | | | | FA1 | FA2 | | |
| 3 | 3 | - | - | 1 | 20 | 20 | 60 | 100 |

**Prior knowledge of:**

1. Networking Layers, IP addressing
2. Distributed Systems

**is essential**

**Course Objectives:**

1. Apply routing algorithms to optimize network path selection and enhance overall network performance.
2. Explore IP traffic engineering principles and routing protocols to evaluate network efficiency.
3. Understand and apply routing principles across diverse domains.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Illustrate routing algorithms to optimize network path selection and performance.
2. Compare different routing protocols used in network communication.
3. Use basic network algorithms for efficient IP lookup and packet classification.
4. Apply traffic engineering principles in IP/MPLS networks.
5. Interpret routing techniques in SDN and NGN networks.
6. Apply routing algorithms to solve problems in various real-world applications.

**Detailed Syllabus**

| Unit | Description | Duration (Hrs) |
|---|---|---|
| 1 | **Network Routing Algorithms: An Introduction**<br>Router and Network Architecture, Bellman–Ford Algorithm and the Distance Vector Approach, Dijkstra's Algorithm: Centralized Approach and Distributed Approach, Shortest Path Computation with Candidate Path Caching, Widest Path Computation with Candidate Path Caching, *k*-Shortest Paths Algorithm. | 7 |
| 2 | **Routing Protocols: Framework and Principles**<br>Interior Gateway Routing Protocol (IGRP): Routing Information Protocol, Open Shortest Path First Protocol, Exterior Gateway Routing Protocol: Border Gateway Protocol, Multicast Routing: Protocol-Independent Multicast (PIM), Distance Vector Multicast Routing Protocol (DVMRP). | 7 |
| 3 | **Network Bottlenecks & Algorithmic thinking:** Network Bottleneck, Network Algorithmics, Strawman solutions, Thinking Algorithmically, Refining the Algorithm, Cleaning up, Characteristics of Network Algorithms<br>**IP Address Lookup Algorithms:** Impact, Address Aggregation, Longest Prefix Matching, Naïve Algorithms, Binary, Multibit and Compressing Multibit Tries, Search by Length Algorithms, Search by Value Approaches, Hardware Algorithms, Comparing Different Approaches<br>**IP Packet Filtering and Classification:** Classification, Classification Algorithms, Naïve Solutions, Two-Dimensional Solutions, Approaches for d Dimensions | 8 |

| | | |
|---|---|---|
| 4 | **Packet Scheduling and Traffic Engineering of IP/MPLS Networks**<br>**Packet Queueing and Scheduling:** First-In, First-Out Queueing, Priority Queueing, Round-Robin and Fair Queueing, Weighted Round-Robin and Weighted Fair Queueing, Queuing models<br>Traffic, Stochasticity, Delay, and Utilization, TCP Throughput and Possible Bottlenecks, Bandwidth-Delay Product, Router Buffer Size<br>**An Architectural Framework, VPN Traffic Engineering, Problem Illustration:** Layer 3 VPN, LSP Path Determination: Constrained Shortest Path Approach, General Modeling Framework, Routing/Traffic Engineering for Voice Over MPLS | 8 |
| 5 | **Intelligent Routing in SDN and NGN Architectures**<br>**Introduction to NGNs and SDNs:** Overview of Next Generation Networks: architecture, goals, and features, Basics of Software-Defined Networking: control plane vs. data plane separation, Key protocols: OpenFlow, NETCONF, Introduction to SDN controllers.<br>**Routing in SDNs:** Centralized routing logic in SDNs, Path computation via SDN controller APIs, Routing Examples: shortest path, multipath, traffic-engineered paths.<br>**Algorithms for Traffic Engineering and QoS in NGNs:** Constraint-based routing (e.g., delay, bandwidth), Multiprotocol Label Switching (MPLS) and Segment Routing, Application-aware routing using machine learning (intro level). | 8 |
| 6 | **Routing Algorithms: Applications and Societal Implications**<br>**Routing in Healthcare Systems:** Routing Algorithms in Telemedicine and Remote Healthcare Services, Role of Robotics in Surgery and Healthcare Automation<br>**Routing in Robotics and Autonomous Vehicles:** Path Planning in Robotics Using Dijkstra's Algorithm, A* Algorithm for Route Optimization, Routing Strategies in Self-Driving Cars<br>**Routing in Transportation and Logistics:** GPS Navigation Systems, Fleet Management and Logistics Routing Strategies, Public Transport System Routing and Scheduling | 7 |
| | **Total** | **45** |

**Text Books:**


1. Medhi D, Ramasamy K. Network routing: algorithms, protocols, and architectures. Morgan kaufmann; 2017 Sep 6.
2. F Kurose J. Computer networking: a top-down approach. Pearson Education; 2021.
3. Stallings W. Data and computer communications. Pearson Education India; 2007.


**Reference Books:**


1. Anderson JM, Nidhi K, Stanley KD, Sorensen P, Samaras C, Oluwatola OA. Autonomous vehicle technology: A guide for policymakers. Rand Corporation; 2014 Jan 10.
2. Andrew S. Tanenbaum & Maarten Van Steen. Distributed Systems: Principles and Paradigms.


**E-Resources:**

1. **https://onlinecourses.nptel.ac.in/noc21_cs18/preview**

| Program: | B. Tech. I.T. | | | | | Semester: | | VI |
|---|---|---|---|---|---|---|---|---|
| Course: | PEC-3 Laboratory: Application of Routing Algorithms Laboratory | | | | | Code: | | BIT26PE12 |
| Credit | Teaching Scheme (Hrs./Week) | | | | Evaluation Scheme and Marks | | | |
| | Lecture | Practical | Tutorial | Other | TW | PR | OR | Total |
| 1 | - | 2 | - | - | 50 | - | - | 50 |

**Prior knowledge of:**

1. Networking
2. Programming

**is essential**

**Course Objectives:**

1. Apply routing algorithms and network protocols to optimize network performance in dynamic environments.

**Course Outcomes:**

After learning the course, the students will be able to:

1. Implement various routing algorithms on various network topologies.
2. Evaluate the impact of packet classification and scheduling on routing efficiency.
3. Apply routing algorithms within SDN environments and for real-world applications.

**Guidelines:**

Continuous assessment of the laboratory is done based on overall performance and laboratory assignments performance of students. Each laboratory assignment assessment will assign marks based on parameters with appropriate weights. Suggested parameters for overall assessment as well as each laboratory assignment assessment include - Neatness and cleanliness in Experimental write-up, Efficient Code, Innovation, Interpretation of results and conclusions, Punctuality, Overall behaviour, attitude and performance.

**Detailed Syllabus**

| Assignment No. | Suggested List of Assignments (Any - 6) |
|---|---|
| 1. | Configure and analyze IP addressing by designing a network with routers, switches, and PCs. Assign IPv4 and IPv6 addresses and implement subnetting. Use traceroute command to examine routing paths in a simulation tool like Cisco Packet Tracer or GNS3. |
| 2. | Implement Bellman–Ford algorithm to examine their performance, convergence time, and suitability for different network topologies. |
| 3. | Implement basic two-dimensional packet classification using source and destination IP ranges and understand filtering algorithms. |
| 4. | Simulate a simplified network with a potential bottleneck and refine the routing or load balancing algorithm to reduce packet loss or delay. |
| 5. | Analysis of Packet Queueing and Scheduling Mechanisms in Routers. Simulate a simple FIFO queue in Python or NS2<br>a. Generate packet arrivals following a Poisson distribution.<br>b. Implement a queue with limited buffer size and observe packet drops.<br>c. Implement a simple Priority Queue using Python.<br>d. Compare the behavior of FIFO vs. Priority Queuing by generating packets with different priority levels. |
| 6. | Configure routers with different buffer sizes, analyze packet drops and delays, and determine the impact of buffer size on overall network performance. |

| | |
|---|---|
| 7. | Simulate a Smart Grid communication network using SDN principles, enabling traffic prioritization for different types of grid data (e.g., control commands, meter readings, video surveillance). |
| 8. | Implement static routing using Ryu SDN controller and Mininet network emulator. |
| 9. | Implement Dijkstra's shortest path algorithm to determine the optimal route between locations in a simulated GPS navigation system. |
| 10. | Study and analyze the Data Center implementation and infrastructure. |
| 11. | Implement A* Algorithm for path planning in robotic navigation and analyze routing strategies used in self-driving cars. |

**Reference Books:**


1. Anderson JM, Nidhi K, Stanley KD, Sorensen P, Samaras C, Oluwatola OA. Autonomous vehicle technology: A guide for policymakers. Rand Corporation; 2014 Jan 10.
2. Andrew S. Tanenbaum & Maarten Van Steen. Distributed Systems: Principles and Paradigms.


**E-Resources:**

1. https://onlinecourses.nptel.ac.in/noc21_cs18/preview.

# Vision and Mission of Information Technology Department

## Department Vision

To become a front-runner in the western region in preparing Information Technology engineers with academic excellence and research skills empowering their roles in technology and society.

## Department Mission

1. To equip students with the skills and knowledge through a dynamic learning environment
2. To collaborate with industries to nurture proficient Information Technology Engineers
3. To cultivate a spirit of research, innovation, and entrepreneurship to address community and business challenges.
4. To imbibe work ethics and leadership skills through co-curricular and extracurricular activities.