

Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus of

T.Y. B. Tech. (Computer Engineering)

Effective from Academic Year 2020-21

Prepared by: - Board of Studies in Computer Engineering

T. Y. B. Tech. Computer Engineering AY 2020-21

Module V

Sr.No.	Subject Code	Subject Name- TY Sem/Module V	Credits Module V	Contact Hours Module V		
				Theory (Hrs/Week)	Lab (Hrs/Week)	Tutorial
S1	CS3201	Computer Network Technology	5	3	2	1
S2	CS3203	System Software	5	3	2	1
S3	CS3202	Artificial Intelligence	5	3	2	1
S4	CS3208	Cyber Security	4	3	2	0
S5	CS3211	Engineering Design-I	1	0	0	2
S6	CS3209	Engineering Design and Innovation-III	4	0	0	0
		Total	24	12	8	5
				25		

Module VI

Sr.No.	Subject Code	Subject Name- TY Sem/Module VI	Credits Module VI	Contact Hours Module VI		
				Theory (Hrs/Week)	Lab (Hrs/Week)	Tutorial
S1	CS3205	Design and Analysis of Algorithms	5	3	2	1
S2	CS3204	Automated Software Engineering	5	3	2	1
S3	CS3206	Data Science	5	3	2	1
S4	CS3207	Compiler Design	4	3	2	0
S5	CS3211	Engineering Design-I	1	0	0	2
S6	CS3209	Engineering Design and Innovation-III	4	0	0	0
		Total	24	12	8	5
				25		

Module V Courses

Syllabus Template**CS3201::Computer Network Technology****Course Prerequisites: Data Communications****Course Objectives:**

1. To understand packet switching architectures of networking reference models.
2. To learn communication framework working for TCP and UDP protocols.
3. To analyze the working of application layer protocols.
4. To learn front end, client side, server side and back end technologies.
5. To develop three tier websites using client side, server side and back end technologies

Credits:..5.....**Teaching Scheme Theory:...3 Hours/Week****Tut: 1 Hours/Week****Lab:...2.Hours/Week**

Course Relevance: The key technology of the information age is communications. Computer network is a truly global area of study, both because the technology enables global communication over telephone lines and the Internet. Computer Networks and web technologies are the backbone of all IT infrastructures and their applications in the world. These technologies and applications often emerge in communication within countries of countries and spread rapidly around the world. Most of the jobs available in the IT industries are web technology related.

SECTION-1**Topics and Contents**

Network Layer: Review of OSI and TCP/IP Models. Functions of network layer, Switching Techniques: Circuit and Packet Switching. Network Layer Protocols: IP Protocol, ICMP Protocol, Logical Addressing: IPv4 and IPv6 is addressing, Subnetting, NAT, CIDR. IP, ICMP, Routing Protocols: Distance Vector, Link State, and Path Vector. Routing on the Internet: RIP, OSPF and BGP. Congestion Control and QoS. Transport Layer: Virtual and Datagram Circuits, Berkeley Sockets, Addressing, Connection Establishment, Connection Release, Flow control and Buffering, Multiplexing. HTH Layer Protocols: TCP, TCP Timer

management, UDP. Quality of Service: TCP Congestion Control. Traffic Shaping: AIMD. Application Layer: Address Resolution: Domain Name System (DNS). WWW: Hyper Text Transfer Protocol (HTTP) and HTTPS with SSL. Web Service. Email: SMTP, MIME, POP3 and Webmail. File Transfer: FTP, Dynamic Logical Addressing: Dynamic Host Control Protocol (DHCP).

SECTION-II

Topics and Contents

Web Development Process, Front End Tools: Introduction to web technology, internet and www, Web site planning and design issues, HTML5: structure of html document, HTML elements: headings, paragraphs, line break, colors& fonts, links, frames, lists, tables, images and forms, , CSS, Bootstrap , XML. **Client Side Technologies:** HTML5 forms Validation, JavaScript: Overview of JavaScript, Data types, Control Structures, Arrays, Functions and Scopes, Objects in JS, DOM: DOM levels, DOM Objects and their properties and methods, Manipulating DOM, JQuery: Introduction to JQuery, Loading JQuery, Selecting elements, changing styles, creating elements, appending elements, removing elements, handling events. Introduction to JSON. **Server Side Technologies:** PHP: Introduction to PHP, Features, sample code, PHP script working, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions, Form handling, Cookies & Sessions, File Handling, Exception Handling, E-mail, MySQL with PHP, AJAX

List of Tutorials: (Any Three)

1) Examples of Network Layer Logical Addressing

- (a) Classful IP and CIDR : Subnetting, IP Prefixes
- (b) NAT Mapping: Public to Private IP and Port Mapping
- (c) Packet Delivery in Internetwork : Packets traversing through different subnetworks with different MTU and Speeds
- (d) Packet Dropping Probabilities of Routers

2) Examples of Network Layer Routing

- (a) Shortest Path and Spanning Tree

- (b)Dijkstra's Algorithm
- (c)Distance Vector Routing
- (d)Link State Routing
- (e)ECMP

3)Examples of Transport Layer

- (a)TCP Connection Establishment: SYN and ACK, Normal Packets
- (b)Flow Control : Calculating Optimal Size of Sliding Window
- (c)Cumulative ACK scheme
- (d)Smoothed RTT
- (e)Slow Start and Additive Increase

4)Examples of Application Layer

- (a)DNS: URL Domain Processing
- (b)Performance of HTTP1.0 and HTTP1.1
- (c)CDN

5)Examples of html and CSS

6)Examples of html form validation

- (a)Email Validation
- (b)Mob No Validation

7) Examples of Bootstrap and XML

8) Examples of JavaScript and JQuery

9) Examples of PHP

10) Examples of PHP and MySQL

List of Practical's: (Any Six)

1)Setting up small computer networks:

Set up a small wired and wireless network of 2 to 4 computers using Hub/Switch/Access point. It includes installation of LAN Cards, Preparation of Cables, Assigning unique IP addresses and sharing C drive on each machine.

2)Installation of Web server:

Installation and configuration web server using Apache Tomcat for Linux. Hands on for network commands - ping, ipconfig, ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute, tracert, nmap and access the web pages from another computer.

3) TCP Socket Programming using Single Thread

Write a program using TCP sockets for following

- a. Say Hello to Each other
- b. File transfer
- c. Calculator

Capture the packets between client and server using Wireshark Packet Analyzer Tool for peer to peer mode.

4) UDP Socket Programming using Single Thread

Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Capture the packets between client and server using Wireshark Packet Analyzer Tool for peer to peer mode.

5) Understanding protocol stack of Intranet

Analyze packet formats of Ethernet, IP, TCP and UDP captured through Wireshark for wired networks.

6)Distance Vector Routing Protocol

Write a program to find the shortest path using Bellman Ford Equation for Distance Vector Routing Protocol which is used by Routing Information Protocol (RIP).

7)Link State Routing Protocol

Write a program to find the shortest path using Dijkstra's Equation for Link State Routing

Protocol which is used by Open Shortest Path First Protocol (OSPF) in the Internet.

8) **Preparation of TCP, IP Packets (Demo Assignment)**

Write a program to prepare TCP and UDP packets using header files and send the packets to the destination machine in peer to peer mode. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

9) Develop a basic web page using the HTML tags you learned in class.

10) Create an admission Template form for VIT admission Process? Perform the validation for email and phone no fields

11) Create an one IT company Template with video in Background (The Web Page must be Responsive and the page contains video in Background)

12) Write a JavaScript program to reverse the elements of a given array.

13) Develop a website using toggleable or dynamic tabs or pills with bootstrap and JQuery

14) Assume we have a file named "webtech.txt", write the correct syntax to open and read the file content.

15) Create three MySQL database tables and write php scripts to read, insert & delete data

16) Write a program to calculate Electricity bill in PHP

(a) You need to write a PHP program to calculate electricity bill using if-else conditions.

(b) Conditions

For first 50 units – Rs. 3.50/unit

For next 100 units – Rs. 4.00/unit

For next 100 units – Rs. 5.20/unit

For units above 250 – Rs. 6.50/unit

You can use conditional statements

List of Projects:

- 1.TCP based Multithreaded HTTP client server
- 2.TCP based Multithreaded FTP client server
- 3.UDP based Multithreaded TFTP client server
- 4.TCP based Multithreaded SMTP and POP3 mail client server
- 5.TCP based Multithreaded Chat client server
- 6.UDP based Multithreaded Chat client server
- 7.UDP based Multithreaded Audio Conferencing client server
- 8.UDP based Multithreaded Video Conferencing client server
- 9.Implementation of RIP/OSPF/BGP using Packet Tracer
- 10.Simulation of AODV routing protocol using NS2/NS3/OMNet
11. Cloud based file sharing platform
12. Develop a Website with NLP as a backend
13. Student Grievance System
14. Workflow Management System for MNC
15. Browser-based Game Website using HTML, CSS, JavaScript, Bootstrap
16. Develop an web application that help to farmers to solve their farming problems
17. GST Billing Software for Small Business
18. Online Crime Reporting System using PHP
19. Develop an Online College Voting System
20. Develop an Online Loan Processing System for Farmers.

List of Course Seminar Topics:

1. IP Addressing using CIDR
2. IP Addressing using IPv6
3. RIP implementation for campus network
4. OSPF implementation in Internet
5. BGP implementation in Internet
6. Simple Network Management Protocol implementation in Internet
7. Evolution of Web Technology
8. CSS and Bootstrap
9. JavaScript and JQuery
10. JSON and AJAX
11. Cookies & Sessions
12. PHP and MySQL

List of Course Group Discussion Topics:

1. Fast Ethernet (Encoding Framing, Modulation, Multiplexing, Diameter etc)
2. Gigabit Ethernet (Encoding Framing, Modulation, Multiplexing, Diameter etc)
3. 10G Ethernet (Encoding Framing, Modulation, Multiplexing, Diameter etc)
4. IEEE802.11b protocol based on HR-DSSS for wireless physical layer standard
5. IEEE 802.11g protocol based on ERP-OFDM for wireless physical layer standard
6. IEEE 802.11n protocol based on HT-OFDM for wireless physical layer standard
7. IEEE 802.11ac protocol based on VHT-OFDM for wireless physical layer standard
8. Evolution of web technologies
- 9.HTML v/s XML
- 10.JavaScript and PHP
11. 21st Century Web Technologies
12. Sql Vs MangoDB
13. Internet Privacy

List of Home Assignments:**Design:**

1. RIP Routing Protocol using packet tracer
- 2.OSPF Routing Protocol using packet tracer
- 3.BGP Routing Protocol using packet tracer
- 4.Front end and Back end Design for Student Grievance System
5. IGRP Protocol

Case Study:

1. RFC for wired TCP
2. RFC for wireless TCP
3. RFC for HTTPS
4. Emerging Web Technology
5. Databases for Servers
6. A case study of JQuery used in any real-time healthcare web application

Blog

1. Internet Logical Addressing
2. Internet Routing Protocols
3. Applications Layer Protocols
4. Front End Technologies
5. Client Side Technologies
6. Server Side Technologies
7. Web Services

Surveys

1. IEEE 802.1 Physical layer standard for Internet
2. IEEE 802.3 Ethernet Standard for LAN
3. IEEE 802.11 Wireless Standard for Wi-Fi
4. Survey on Front End Technologies
5. Survey on Client Side Technologies
6. Survey on Server Side Technologies

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

MSE(15)+ESE(15)+HA(10)+LAB(10)+CP(10)+CVV(20)+SEMINAR(10)+GD(10)

Text Books: (As per IEEE format)

1. Andrew S. Tenenbaum, "Computer Networks", 5th Edition, PHI, ISBN 81-203-2175-8.
2. Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet", Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204
3. Achyut Godbole & Atul Kahate, "Web Technologies: TCP/IP to Internet Application Architectures", McGraw Hill Education publications, Third Edition, 2016
4. Ralph Moseley & M. T. Savaliya, "Developing Web Applications", Wiley publications, Second Edition, 2014

Reference Books: (As per IEEE format)

1. Fourauzan B., "Data Communications and Networking", 5th edition, Tata McGraw- Hill, Publications, 2006
2. Matthew S. Gast "802.11 Wireless Networks", O'Reilly publications; 2nd Edition.
3. C. Siva Ram Murthy and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols" Prentice Hall, 2004
4. Holger Karl and Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks", Wiley, ISBN: 0-470-09510-5
5. Adam Bretz & Colin J Ihrig, "Full Stack Javascript Development with MEAN", SPD, First Edition 2015, Indian Reprint September 2015
6. Giulio Zambon, "Beginning JSP, JSF and Tomcat", Apress Publication, Second Edition, 2013
7. Jeremy McPeak & Paul Wilton, "Beginning JavaScript", Wrox Publication, Fifth Edition, 2015

Moocs Links and additional reading material:

www.nptelvideos.in

w3schools.com

www.coursera.com

Course Outcomes:

1. Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols
2. Demonstrate sustainable engineering practice indicating the scientific purpose and utility of communication frameworks and standards.
3. Develop Client-Server architectures and prototypes by the means of correct standards, protocols and technologies.
4. Understanding web technology architectures for TCP/IP service networks.
5. Design of small case websites using front end, server side and backend technologies.
6. Development of enterprise websites web sites using client side frameworks, server side and backend technologies.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO6	PO11	PSO3
2	3	3	2	1	3

CO attainment levels

Attainment Levels: 1, 5, 3, 4, 2, 4

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

High Performance Networks, Wireless Networks, Mobile Networks, IoT

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Web Developer, IT Engineer, Network Administrator

Syllabus Template

CS3203::System Software

Course Prerequisites: Computer Organization, Data Structure

Course Objectives:

1. Understand structure & function of Operating System
2. Learn CPU scheduling algorithms
3. Deal with deadlock
4. Learn memory management techniques.
5. Understand different system software's & their functionalities.
6. Design device drivers for Linux.

Credits:5.....

Teaching Scheme Theory:...3 Hours/Week

Tut: 1 Hours/Week

Lab:...2.Hours/Week

Course Relevance: Systems software are programs that manage the resources of the computer system and simplify applications programming. They include software such as the operating system, database management systems, networking software, translators, and software utilities. This course is considered as core subject as it supports the hardware abstraction to users.

SECTION-1

Introduction to OS: What is OS, functions of OS, OS Services, System Calls, Types of system calls Types of OS, Structures of OS: Monolithic, Layered, Virtualization

Processes: Process Concept, Process States: 2, 5, 7 state models, Process Description, Process Control. Threads: Multithreading models, Thread implementations, threads, Symmetric Multiprocessing. Concurrency: Issues with concurrency, Principles of Concurrency Mutual Exclusion: H/W approaches, S/W approach, OS/Programming support: Semaphores, Mutex and Monitors. Readers-Writers problem, Producer Consumer problem, Dining Philosopher problem.

Process Scheduling: Uni-processor Scheduling, Scheduling: Preemptive, Non preemptive, Long-term, Medium-term, Short term scheduling. Scheduling Algorithms: FCFS, SJF, RR,

Priority.

Deadlocks: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery

SECTION-11

Memory Management: Memory Management concepts: Memory Management requirements, Memory Partitioning: Fixed, Dynamic Partitioning, Buddy Systems Fragmentation, Paging, Segmentation, Address translation.

Placement Strategies: First Fit, Best Fit, Next Fit and Worst Fit.

Virtual Memory: Concepts, Swapping, VM with Paging, Page Table Structure, Inverted Page Table, Translation Lookaside Buffer, Page Size, VM with Segmentation, VM with Combined paging and segmentation.

Page Replacement Policies: FIFO, LRU, Optimal, Clock.

Introduction to System Software: Introduction, software types, software hierarchy, components of system software, machine structure, interfaces, address space, levels of system software, recent trends in software development. Language processors: Programming languages and language processors, fundamentals of language processing, life cycle of a source program, language processing activities, data structures for language processing: search data structures, allocation data structures.

Introduction to Device Driver: Requirements of Device Drivers, Role of Device Drivers, Classes of Devices, Security issues, Design issues, PCI Bus Drivers, The Peripheral Component Interconnect (PCI) Interface, Peripheral component interconnect (PCI) addressing, Boot time, Configuration registers and initialization, Linux PCI initialization, Old-style PCI probing, USB Drivers.

List of Tutorials: (Any Three)

- 1) Linux simple & advanced commands
- 2) Shell Script
- 3) Classical Problems of Synchronization
- 4) CPU Scheduling Algorithms
- 5) Deadlock Prevention, Avoidance, detection algorithms
- 6) Memory Cache Mapping techniques
- 7) Memory placement strategies
- 8) Memory Page replacement algorithms
- 9) Disk Scheduling Algorithms
- 10) Device Drivers

List of Practical's: (Any Six)

- 1) Write a shell script.
- 2) Solve Classical Problems of Synchronization using mutex & semaphore.
- 3) Implementation of CPU Scheduling Algorithms.
- 4) Implementation of Deadlock Prevention, Avoidance, detection algorithms
- 5) Implementation of Memory placement strategies – best fit, first fit, next fit & worst fit.
- 6) Implementation of Page replacement algorithms
- 7) Implementation of Disk Scheduling Algorithms
- 8) Write a device driver for char device
- 9) Write a device driver for block device
- 10) Write a device driver for network device**

List of Projects:

- 1.. Design and implementation of a multi programming operating system phase 1
 - i. CPU/ Machine Simulation
 - ii. Supervisor Call through interrupt
2. Design and implementation of a Multiprogramming Operating System: Stage II
 - i. Paging
 - ii. Error Handling
 - iii. Interrupt Generation and Servicing
 - iv. Process Data Struc
3. Design and implementation of a Multiprogramming Operating System: Stage III
 - i. I/O Channels
 - ii. Multiprogramming
 - iii. I/O Spooling
4. Design multi programming operating system phase 1 with arithmetic & logical instruction
- 5.Design multiprogramming operating system phase 3 without swapping
6. Design multi programming operating system phase 3 with swapping
7. Write a linux character device driver module that implements the open(), close(), read() and write() system calls for a character device.
8. Write a device driver on linux system.
9. Design a kernel module for linux/ unix
10. Design & Implementation of DLL on Linux shared library.

List of Course Seminar Topics:

1. Android Operating System
2. Paging & Segmentation in Windows OS
3. Linux Architecture
4. Linux Administration
5. Windows Administration
6. Android Administration
7. Device Administration in Windows 10
8. System Calls
9. DLLs in Linux
- 10. DLLs in Windows**

List of Course Group Discussion Topics:

- 1.Types of operating systems
2. Structure & function of OS
3. Types of CPU scheduling algorithms
4. Desktop OS & mobile OS
5. Memory management in Desktop OS & mobile OS
6. Process management in Desktop OS & mobile OS
7. Device management in Desktop OS & mobile OS
8. Deadlock recovery techniques
9. Deadlock prevention & avoidance strategies
10. Deadlock avoidance & detection strategies
11. File system in desktop OS & mobile OS.

List of Home Assignments:**Design:**

1. I/O Management
2. File Management
3. Disk Scheduling
4. File Management
5. File System

Case Study:

1. Process Management in Linux
2. Process Management in Android
3. Memory Management in Linux
4. Memory Management in Android
5. Process Management in Windows

Blog

1. File System of Windows
2. File System of Linux
3. File System of Android
4. File System of iOS
5. Memory Management in Windows

Surveys

1. Mobile OS used in Smart Phones
2. OS used in data centers
3. Distributed OS & applications
4. Device Drivers for various devices

Suggest an assessment Scheme:

MSE(15)+ESE(15)+HA(10)+LAB(10)+CP(10)+CVV(20)+SEMINAR(10)+GD(10)

Text Books: (As per IEEE format)

1. Stalling William; "Operating Systems", 6th Edition, Pearson Education.
2. Silberschatz A., Galvin P., Gagne G.; "Operating System Concepts", 9th Edition, John Wiley and Sons.
3. D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw Hill Publications, ISBN – 0074635794
4. John J Donovan; " Systems Programming " ; Tata Mc-Graw Hill edition , ISBN-13978-0-07-460482-3

Reference Books: (As per IEEE format)

1. Silberschatz A., Galvin P., Gagne G ; "Operating System Principles" 7th Edition John Wiley and Sons.
2. Yashavant Kanetkar; "Unix Shell Programming", 2nd Edition, BPB Publications.
3. Forouzan B. A., Gilberg R. F.; "Unix And Shell Programming", 1st Edition, Australia Thomson Brooks Cole.
4. Achyut S. Godbole , Atul Kahate; "Operating Systems", 3rd Edition, McGraw Hill.
5. Robert Love, " Linux System Programming " ;O'Reilly, ISBN 978-0-596-00958-8
6. Mahesh Jadhav; " Easy Linux Device Driver "; HighTechEasy publishing, Second edition.
7. Ray Duncan; "Advanced MSDOS programming"; Microsoft press

Moocs Links and additional reading material: www.nptelvideos.in

<https://nptel.ac.in/courses/106/105/106105214/>

Course Outcomes:

- 1) Examine the functions of a contemporary Operating system with respect to convenience, efficiency and the ability to evolve.
- 2) Demonstrate knowledge in applying system software and tools available in modern operating system.
- 3) Apply various CPU scheduling algorithms to construct solutions to real world problems.
- 4) Identify the mechanisms to deal with Deadlock.
- 5) Understand the organization of memory and memory management.
- 6) Discriminate among different System software and their functionalities.

CO PO Map

CO1- PO2(2)

CO2-PO3(3)

CO3-PO4(3)

CO4- PO10(2)

CO5-PO12(1)

CO6-PSO3(3)

CO attainment levels

CO1-5

CO2-2

CO3-3

CO4-2

CO5-1

CO6-4

Future Courses Mapping:

Distributed Operating Systems

High Performance Computing

Distributed Computing

Compiler Design

Job Mapping:

System Administrator (Linux)

Quality Engineer

Operating System Analyst

Data Developer

CS3202 :: ARTIFICIAL INTELLIGENCE**Course Prerequisites:** Mathematics, Data Structures**Course Objectives:**

- 1 To introduce the concepts, techniques and building blocks of Artificial Intelligence.
- 2 To evaluate the searching Techniques and its implementation
- 3 To Analyze planning techniques and its applications in developing solutions to real-world problems.
- 4 To apply information gained through knowledge representation and uncertainty to a given situation
- 5 To generate an ability to design, analyze and perform experiments on real life problems using various AI Techniques.
- 6 To build artificial intelligence models and implement in real life scenario.

Credits:..5.....**Teaching Scheme Theory:...3 Hours/Week****Tut: 1 Hours/Week****Lab:...2.Hours/Week**

Course Relevance: Artificial intelligence (AI) is currently one of the hottest buzzwords in tech and with good reason. The last few years have seen several techniques that have previously been in the realm of science fiction slowly transformed into reality. The importance of Artificial Intelligence has been increasing as a growing number of companies are using these technologies to improve their products and services, evaluate their business models, and enhance their decision-making process.

SECTION-1

Topics and Contents

Fundamentals of Artificial Intelligence: Introduction, A.I. Representation, Non-AI and AI Techniques, State Space Search, Production Systems, Problem Characteristics, types of production systems, Intelligent Agents and Environments, concept of rationality, structure of agents, problem solving agents, and problem formulation.

Searching: Depth First Search, Breadth First Search, Generate and test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Means-Ends Analysis. Game playing: Minimax Search, Alpha-Beta Cutoffs.

Planning: Blocks world, STRIPS, Implementation using goal stack, Partial Order Planning, Hierarchical planning, and least commitment strategy. Conditional Planning, Continuous Planning.

SECTION-II

Topics and Contents

Knowledge Representation: Knowledge based agents, Wumpus world, Propositional Logic: Representation, Inference, Reasoning Patterns, Resolution, First order Logic: Representation, Reasoning Patterns, Forward and Backward Chaining. Basics of PROLOG: Representation, Structure, Backtracking, Expert System, MYCIN, DART, ZOOM,

Uncertainty: Non Monotonic Reasoning, Logics for Non Monotonic Reasoning, Forward rules and Backward rules, Justification based Truth Maintenance Systems, Semantic Nets Statistical Reasoning, Markov Networks.

List of Tutorials: (Any Three)

- 1) Intelligent Agents and Environments
- 2) Uninformed searching Techniques
- 3) Informed searching Techniques
- 4) Gaming
- 5) Planning
- 6) Blocks world
- 7) STRIPS
- 7) Wumpus world
- 8) Propositional Logic
- 9) Expert System
- 10) Chat bots and Robotics

List of Practical's: (Any Six)

1. Tic-Tac-Toe Game playing
2. Analysis of AI and Non-AI technique by implementing any two player game.
3. Implement Uninformed (Exhaustive) searching Technique/s
4. Implement Informed (Heuristic) searching Technique/s
5. Implement steepest ascent hill climbing for 8-puzzle/other application
6. Wumpus world
7. Propositional Logic
8. Expert System
9. Chat bots and Robotics

10. Implement simple hill climbing for 8-puzzle / other application
11. Analysis of Constraint satisfaction Problems.
12. Implement Robotic hand
13. Implement Perception model for detection

List of Projects:

- 1 Inventory management E Commerce
- 2 stock market price prediction
- 3 Identification / detection
- 4 Product Delivery Drones
- 5 Pick and drop robotic arm
- 6 Arrangement of blocks
- 7 Smart city water / light management system
- 8 Human Tracking system
- 9 Automatic Interview Conduction system
- 10 Student Information Chatbot Project.
11. Product Review Analysis For Genuine Rating.
- 12 Customer Targeted E-Commerce
- 13 College Enquiry Chat Bot
- 14 Artificial Intelligence HealthCare Chatbot System
- 15 Intelligent Tourist System Project

List of Course Seminar Topics:

- 1) Basic Concepts Artificial Intelligence
- 2) Intelligent Agents
- 3) Uninformed searching Techniques
- 3) Informed searching Techniques
- 4) Gaming Techniques
- 5) Planning Techniques
- 6) Applications of AI
- 7) Wumpus world
- 8) Propositional Logic
- 9) Expert System
- 10) Chat bots
- 11) AI Robots

List of Course Group Discussion Topics:

1. Artificial Intelligence and Machine Learning
2. Artificial Intelligence and Data science
3. Artificial Intelligence applications
4. Artificial Intelligence future
5. Artificial Intelligence after 10 years / 2030
6. Uninformed searching and Informed searching Techniques
7. Chatbots and Recommender systems

8. Will Automation and AI Reduce or Increase Jobs.

9. Cashless Economy using AI

10. AI in covid-19 situations

List of Home Assignments:

Design:

- 1 AI Accessibility Design
- 2 Best Artificial Intelligence Design for any one application
- 3 AI Customer Experience Design
- 4 AI Data-Informed Design
- 5 AI Decision Architecture
- 6 AI application Designing for Children
- 7 AI Designing for Senior Citizens
- 8 AI in eCommerce Design
- 9 AI Enterprise UX Design
- 10 AI Experience Design

Case Study:

1. How Mercedes Is Preparing For The 4th Industrial Revolution using AI
2. How Indian Retail Giant Is Using AI And Robots To Prepare For The 4th Industrial Revolution
3. Rolls-Royce And Google Partner To Create Smarter, Autonomous Ships Based On AI
4. The Amazing Ways Tesla Is Using Artificial Intelligence And Big Data
5. The Incredible Ways John Deere Is Using Artificial Intelligence To Transform Farming

Blog

1. AI Trends
2. AI Research
3. AI Chatbot
4. Chatbot Magazine
5. AI Medical / Agriculture

Surveys

1. Adaption of AI in 2020
2. AI in Industry
3. AI in Digital Marketing
4. AI in Gaming
5. AI after Covid-19

Suggest an assessment Scheme:

HA, PPT, GD, MSE, ESE, LAB, CVV

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

Text Books: (As per IEEE format)

1. Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill
2. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.
3. Deepak Khemani: "A First Course in Artificial Intelligence", Mc Graw Hill
4. Saroj Kaushik: "Artificial Intelligence" Cengage Publication

Reference Books: (As per IEEE format)

1. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley, 1990.
2. Eugene, Charniak, Drew McDermott: "Introduction to Artificial Intelligence.", Addison Wesley
3. Patterson: "Introduction to AI and Expert Systems", PHI
4. Nilsson : "Principles of Artificial Intelligence", Morgan Kaufmann.
5. Carl Townsend, "Introduction to turbo Prolog", Paperback, 1987

Moocs Links and additional reading material: www.nptelvideos.in

www.eduplus.in

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
2. Formalize a given problem in the language/framework of different AI methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, as a Markov decision process, etc)
3. Implement basic AI algorithms (e.g., standard search algorithms or dynamic programming).
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports.
5. Use various symbolic knowledge representations to specify domains and reasoning tasks of a situated software agent.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO4	PO5	PO6	PO10	PSO3
2	3	3	2	1	3

CO attainment levels

CO1-3 CO2-3 CO3-1 CO4-2 CO5-2 CO6-2

Future Courses Mapping:

Artificial Neural Networks, Machine Learning, Soft Computing, Deep Learning.

Job Mapping:

AI Scientist, AI Developer, AI Designer,

Syllabus Template
CS3208::Cyber Security

Course Prerequisites: Computer Networks**Course Objectives:**

1. Learners must be able to discover the programming bugs that will be malicious code, they also must be able to explain various attacks and resolve the bugs to mitigate the treats.
2. Learners must be able to apply various cryptographic techniques to secure the systems developed.
3. Discover and explain various authentication and authorization methods with the access control
4. Articulate the use of various standard security protocols for the layered architecture.
5. Articulate the urgent need for cyber security in critical computer systems, networks, and World Wide Web, and explain various threat scenarios
6. Articulate the cyber threats to critical infrastructures

Credits:4.....**Teaching Scheme Theory:...3 Hours/Week****Tut: Hours/Week****Lab:...2.Hours/Week**

Course Relevance: The increased use of digital technologies and interconnection of these devices has widely increased. This increase of internet use is leading towards the frouds on the internet due to malicious intentions. It is need of time to understand the various types of attacks, attack goals and reasons for the attack such as vulnerability etc. Also it is also important to know the various ways by which information can be secured at the time of transmission. It is very important to have in- depth knowledge about the authentication of the users on the system developed, authorization is also important. As the crime increases the security expert must be able to know the crime has happened or not, how to detect it. What are the various application layer issues for security? This gives the scope of relevance of the study of this course.

SECTION-1

Introduction to Attacks and Hacking: Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack and attack types, Avoiding attacks, Security services. Trustworthiness, Ethical issues and practices, Tradeoffs of balancing key security properties - Confidentiality, Integrity, Availability. Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing, Pharming attack, Dictionary Attacks. Software vulnerabilities: Phishing, buffer overflow, Cross-site scripting attack, Virus and Worm Features, Trojan horse, Social engineering attacks, ransomware, SYN-Flooding, SQL- injection, DNS poisoning, Sniffing, Introduction to Ethical Hacking, Anonymity, Information Gathering, Scanning Networks, Vulnerability Analysis, Operating System Hacking, Hacking Wireless Networks,

Cloud Hacking, IoT Hacking,

Cryptography:

Private key cryptography: Mathematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field). Role of random numbers in security, Importance of prime numbers

Data Encryption Standard: Block cipher, Stream cipher, Feistel structure, round function, block cipher modes of operation, S-DES, Attacks on DES, S-AES, AES.

Public key cryptography: RSA: RSA algorithm, Key generation in RSA, attacks on RSA. Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack. Elliptic Curve Cryptography (ECC): Elliptic Curve over real numbers, Elliptic Curve over \mathbb{Z}_p , Elliptic Curve arithmetic. Diffie-Hellman key exchange using ECC. Chinese remainder theorem.

SECTION-II

Authentication and Authorization: Network Access Control: Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port-Based Network Access Control, SHA-512, Kerberos, X.509 authentication service

IP Security, Session Management, Web Security, Database Security, File Security, Mobile Security

Transport-Level Security: Web Security Considerations, Secure Sockets Layer, Transport Layer Security, HTTPS standard, Secure Shell (SSH) application

Email security: PGP and SMIME

Introduction to Digital Forensics, Data Recovery and OS Forensics, Email Crimes and Violations,

Cyber Security: Cyber Attack, Cyber Reconnaissance, Crimes in Cyber Space-Global Trends & classification, e-commerce security, Computer forensics, facebook forensic, mobile forensic, cyber forensic, digital forensic, Introduction to MQTT and CoAP for IoT.

List of Practical's: (Any Six)

- 1) Acquisition of System Information/ RAM/Volume Shadow Copy/Detecting Encryption in information.
- 2) Forensic of Disc Image/ Registry/ Meta data/ RAM
- 3) Simplified DES implementation
- 4) Simplified AES implementation
- 5) Encryption and Decryption by RSA algorithm
- 6) Implementation of ECC over Diffie Hellman Key Exchange Protocol
- 7) Implementation of authentication algorithms
- 8) Digital forensic of images
- 9) Forensics of Video alteration
- 10) Vulnerabilities finding in Mobile/ computer/ digital devices

List of Projects:

1. Design a System to develop a analyzer which will differentiate between different vulnerability and packets entered using it. This system will detect the intrusions coming through the vulnerabilities.
2. Securing Video Conferencing App for online meetings
3. Steganography for Image/Video/Files
4. Secure Image display on online social media.
5. Secure transfer of government subsidies to farmers/BPL people/ students etc
6. Authentication of users for various applications for integrity, availability, confidentiality.
7. Implementing a system for detecting the modification of videos/images on social media
8. Secure App for online exams detecting Keystroke and camera movements.
9. A system to detect the difference between the voice edited in the audio/video
10. A System to check the vulnerabilities in the websites.

List of Course Seminar Topics:

1. Blockchain architecture and its implementation
2. Cloud Security
3. Mobile Security
4. IoT and Security Issues/ Security Models for IoT
5. Darkweb
6. Docker Security
7. Access control methods for online social media and various organizations
8. Security of Android Vs IOS
9. Machine learning and SCADA Security
10. Security Applications for Smart Cities

List of Course Group Discussion Topics:

1. Security Issues in Android and IOS devices
2. Industry 4.0 and security
3. Blockchain and E-voting system
4. Security of Aadhar Card and other digital cards
5. Automated Home Appliances and Security
6. Programming Bugs and Malicious code in information security
7. Indian Cyber laws and Deficiencies
8. Social Media and Cyber Security
9. Child abuse on online social media and security
10. Need of cyber crime and security in school education.

List of Home Assignments:

Design:

1. Design a secure system using cryptography techniques for security of multimedia files.
2. Design a secure system using steganography for hiding data files in image/video
3. Design a system for educational institutes using authentication and authorization techniques, also give details about the access control policies that must be implemented for the design of system by various places.
4. Design a secure system using SSL/TLS/IPSec for the various organizations
5. Design a system for the analysis of cyber crime using various cyber forensic techniques and compare each technique with respect to integrity, confidentiality, availability

Case Study:

1. How to improve the security of social media? Write a detail case study
2. Find out the vulnerability issues in educational institutes websites/online systems and give solutions to these problem. Perform a detailed case study of the various issues.
3. Write a detail case study about the banking security flows and solutions to these flows.
4. Give a detail case study of the antivirus system giving the flows and solutions to it.
5. Perform the detail case study of various operating systems used for mobile devices and give a secure solution to one for widely used OS.

Blog

1. Dark Web
2. Crypto currency and Economy
3. Cyber crime and solutions
4. Authentication and Access control for social media
5. Cyber forensic and Cyber laws

Surveys

1. Survey on various blockchain related issues/ cryptocurrency/ application systems developed using blockchain
2. Survey on various authentication and access control methods for different applications
3. Steganography and Biometric Systems for authentication
4. Survey of various attacks and its effect on Indian economy and its analysis
5. Problems over Integer Lattices: A Study

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

HA assessment:

Six home assignments will be given one for biweekly. Best of five will be considered for assessment, 20 marks each. (10 marks weightage)

Lab Assessment: 10 marks for continuous assessment of lab assignments

Course project: 15 marks

Technical seminar : 10 marks

Group Discussion: 10 marks

MSE : 15 marks

ESE:15 marks

CVV:15 marks

Text Books: (As per IEEE format)

1. William Stallings; “Cryptography and Network Security-Principles and Practices” 6th Edition , Pearson Education, 2014, ISBN13:9780133354690.
2. Bernard Menezes, “Network Security and Cryptography”, 1st Edition, Cengage Learning, 2010, ISBN 81-315-1349-1.
3. Raef Meeuwisse, “Cybersecurity for Beginners”, 2nd Edition, Cyber Simplicity, 2017, ISBN- 9781911452157

Reference Books: (As per IEEE format)

1. M. Speciner, R. Perlman, C. Kaufman, “Network Security: Private Communications in a Public World”, Prentice Hall, 2002
2. Michael Gregg, “The Network Security Test Lab: A Step-By-Step Guide”, Dreamtech Press, 2015, ISBN-10:8126558148, ISBN-13: 978-8126558148.
3. Matt Bishop, “Computer Security: Art and Science”, 1st Edition, Pearson Education, 2002, ISBN 0201440997.
4. Charlie Kaufman, Radia Perlman and Mike Spencer, “Network security, private communication in a public world”, 2nd Edition, Prentice Hall, 2002, ISBN 9780130460196.
5. V.K. Pachghare, “Cryptography and Information Security”, 2nd Edition, PHI, 2015, ISBN-978-81-203-5082-3.

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

- 1) . Analyze cryptographic techniques using a mathematical approach by examining nature of attack.
- 2) Identify and establish different attacks on the system
- 3) Justify various methods of authentication and access control for application of technologies to various sections of industry and society.
- 4) Design a secure system for protection from the various attacks for 7 layer model
by determining the need of security from various departments of an organization
- 5) Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
- 6) Analyze various types of cyber crime by detecting the crime

CO PO Map

Programme Outcomes												Program Specific Outcome			
PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
	2	3		3	2			1						3	

CO attainment levels

CO	Attainment level
1	4
2	3
3	5
4	3
5	2
6	

Future Courses Mapping:

Blockchain Technologies

Ethical Hacking and Network Defense

Cloud and Data Security

Job Mapping:

Security Engineer/Network Security Engineer

Information Security Analyst

Cyber Security Analyst

Cyber Security Associate

Manager-Information Security Services

Security Consultant

Penetration Testing Engineer

Syllabus Template

CCS3211: : Engineering Design

Course Prerequisites: Problem Based Solving

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. to develop an ecosystem to promote entrepreneurship and research culture among the students.

Credits:.1.....

Teaching Scheme Theory:... Hours/Week

Tut: 2 Hours/Week

Lab:....Hours/Week

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guidelines mentioned herewith. For all courses of ED laboratory course contents of “Engineering Design” are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards , Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Topic and Content: Research in computational biology using CPython

Ecology, Next-generation Sequencing, Clinical Oncology Next-generation Sequencing, Sanger sequencing for DNA, Next-generation sequencing (NGS) for DNA, Illumina next-generation sequencing (INGS) method is based on sequencing-by-synthesis (SBS), and reversible dye-terminators for DNA, genomic DNA, Polymerase chain reaction (PCR), library construction, Southern blotting Southern blotting laboratory technique to detect a specific DNA sequence in a blood or tissue sample., hybridizations, Single nucleotide polymorphisms (SNP), SNP analysis, Molecular diagnostic assays, Sequence comparison and phylogenetic reconstruction, Complex analysis with proteomics data.

List of Practical's: (Any Six)

- 1) Study of protein, population genetics, big genomics dataset
- 2) Biopython – Hand's on to use bioinformatics library of CPython

- 3) PyVCF – Use of bio library for Next-generation Sequencing processing and analysis
- 4) Pysam – Use bio library for SAM and BAM processing and analysis
- 5) HTSeq – Use of bio library for Gnome NGS processing and analysis
- 6) SimuPop – Use of bio library for population genetics simulation
- 7) DendroPY – Use of bio library for proteomics processing and analysis

...not limited to.....**Faculty and students are free to include other area which meets the society requirements at large.**

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

To focus on the higher levels of the Bloom's Taxonomy analyze, apply, evaluate and create.

Text Books: (As per IEEE format)

1. Name(s) of author(s); Title of the book; Edition No., Publisher

2

3

4

Reference Books: (As per IEEE format)

1. Name(s) of author(s); Title of the book; Edition No., Publisher

2

3

4

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO4	PO5	PO7	PO12	PSO1
2	3	3	2	1	3

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

Syllabus Template

CS3209::Engineering Design and Innovation-III**Course Prerequisites:** Problem Based Learning**Course Objectives:**

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Credits:.4.....**Teaching Scheme Theory:... Hours/Week****Tut: Hours/Week****Lab:...Hours/Week**

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of “Trends in Engineering Technology” are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.
- It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:

Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.

This is the sample list to start with. Faculty and students are free to include other area which meets the society requirements at large.

The model begins with the identifying of a problem, often growing out of a question or “wondering”. This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.

- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students’ wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an

interdisciplinary approach in both the analysis and solving phases.

- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?

- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use following mechanism to maintain the track of moving towards the solution.
- How effective is? How strong is the evidence for? How clear is?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDI Sample Case Studies : -

- 1) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis
- 2) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for SAM and BAM processing and analysis
- 3) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for Gnome NGS processing and analysis
- 4) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for population genetics simulation
- 5) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for proteomics processing and analysis

...not limited to.....Faculty and students are free to include other area which meets the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Higher levels of the Booms Taxonomy - analyze, apply, evaluate and create.

Text Books: (As per IEEE format)

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. De Graaff E, Kolmos A., red.: *Management of change: Implementation of problem-based and project-based learning in engineering*. Rotterdam: Sense Publishers. 2007.
2. *Project management core textbook, second edition, Indian Edition*, by Gopalan.
3. *The Art of Agile Development*. By James Shore & Shane Warden.

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO4	PO5	PO7	PO12	PSO1
2	3	3	2	1	3

CO attainment levels

CO1 -4 CO2 -2 CO3-4 CO4-5 CO5 -1 CO6-3

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

Module VI courses

Syllabus Template**CS3205::Design and Analysis of Algorithms**

Course Prerequisites: Basic courses on programming, data structures, discrete structures, theory of computing.

Course Objectives:

1. Students will gain understanding of asymptotic notations and will be able to apply suitable mathematical techniques to find asymptotic time and space complexities of algorithms.
2. Students will develop the ability to formulate computational problems in the abstract and mathematically precise manner.
3. Student will gain understanding of different algorithm design paradigms such as divide and conquer, dynamic programming, greedy, backtracking and will apply suitable paradigm for designing algorithms for computational problems
4. Students will develop understanding of notions of NP-hardness and NP-completeness and their relationship with the intractability of decision problems.
5. Students will design randomized, approximation algorithms for some computational problems.

Credits:..5.....

Teaching Scheme Theory:...3 Hours/Week

Tut: 1 Hours/Week

Lab:...2.Hours/Week

Course Relevance: This is a foundational course for Computer science and Engineering. This course develops algorithmic thinking capability of students. Designing algorithms using suitable paradigm and analysing the algorithms for computational problems has a high relevance in all domains where computer science plays a crucial role (equally in Industry as well as research). This course is also an essential pre-requisite for advanced domain specific algorithmic courses such as Algorithmic Graph Theory, Algorithmic Number Theory, Computational Geometry, Motion planning and Robotics, etc, to give a few examples.

Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic study of any other domain (in computer science or otherwise) which demands logical thinking.

This course is also relevant for students who want to pursue research career in theory of computing, computational complexity theory, advanced algorithmic research.

SECTION-1

Basic introduction and time and space complexity analysis:

Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst-case time and space complexity of algorithms. Overview of searching, sorting algorithms. Adversary lower bounds (for the comparison-based sorting algorithms, for finding second minima). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master's theorem and applications. Proving correctness of algorithms.

Divide and Conquer: General strategy, Binary search and applications, Analyzing Quick sort, Merge sort, Counting Inversions, Finding a majority element, Order statistics (randomized and deterministic algorithms), Josephus problem using recurrence, Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast exponentiation).

Dynamic Programming: General strategy, simple dynamic programming based algorithms to compute Fibonacci numbers, binomial coefficients, Matrix Chain multiplication, Optimal binary search tree (OBST) construction, Coin change problem, 0-1 Knapsack, Traveling Salesperson Problem, All pair shortest path algorithm, Longest increasing subsequence problem, Largest independent set for trees.

SECTION-1I

Greedy and Backtracking strategy:

Greedy: General strategy, Analysis and correctness proof of minimum spanning tree and shortest path algorithms, fractional knapsack problem, Huffman coding, conflict free scheduling.

Backtracking: General strategy, n-queen problem, backtracking strategy for some NP-complete problems (e.g. graph coloring, subset sum problem, SUDOKU)

Introduction to complexity classes and NP-completeness:

Complexity classes P, NP, coNP, and their interrelation, Notion of polynomial time many one reductions reduction, Notion of NP-hardness and NP-completeness, Cook-Levin theorem and implication to P versus NP question, NP-hardness of halting problem. NP-Complete problems (some selected examples from - Satisfiability problem, Circuit-SAT, 3-CNF SAT, vertex cover problem, independent set problem, clique problem, Hamiltonian-circuit problem, subset sum problem, Integer Linear Programming.), reducing NP problems to Integer Linear Programming.

Introduction to Randomized and Approximation algorithms:

Introduction to randomness in computation, Las-Vegas and Monte-Carlo algorithms, Abundance of witnesses/solutions and application of randomization, solving SAT for formulas with “many” satisfying assignments, randomized quick sort, Las-Vegas and Monte-Carlo algorithms for majority search, Karger’s Min-cut algorithm, coupon collector problem, randomized data structures (randomized BST, skip lists)

Introduction to Approximation algorithms for NP-optimization problems, Approximation algorithm for Vertex Cover, metric Traveling-Sales-Person Problem (metric-TSP), Hardness of approximation for TSP.

List of Tutorials: (Any Three)

- 1) Problem solving based on asymptotic notations, solution of recurrences
- 2) Problem solving based on Divide and Conquer strategy
- 3) Advanced problem solving based on Divide and Conquer strategy
- 4) Problem solving based on Dynamic Programming strategy
- 5) Advanced problem solving based on Dynamic Programming strategy
- 6) Problem solving based on Greedy strategy
- 7) Problem solving based on Backtracking strategy
- 8) Proving correctness of algorithms: some techniques
- 9) Adversary lower bound technique
- 10) Problem solving based on complexity classes, NP-completeness.
- 11) Randomized Algorithms
- 12) Approximation Algorithms

List of Practical's: (Any Six)

- 1) Assignment based on some simple coding problems on numbers, graphs, matrices
- 2) Assignment based on analysis of quick sort (deterministic and randomized variant)
- 3) Assignment based on Divide and Conquer strategy (e.g. majority element search, finding kth rank element in an array)
- 4) Assignment based on Divide and Conquer strategy (e.g. efficient algorithm for Josephus problem using recurrence relations, fast modular exponentiation)
- 5) Assignment based on Dynamic Programming strategy (e.g. Matrix chain multiplication, Longest increasing subsequence)
- 6) Assignment based on Dynamic Programming strategy (e.g. All pair shortest path, Traveling Sales Person problem)
- 7) Assignment based on Greedy strategy (e.g. Huffman encoding)
- 8) Assignment based on Backtracking (e.g. graph coloring, n-queen problem)
- 9) Assignment based on Las-Vegas and Monte-Carlo algorithm for majority element search
- 10) Assignment based on factor-2 approximation algorithm for metric-TSP

List of Projects:

1. Applications of A* algorithm in gaming
2. Pac-Man game
3. File compression techniques
4. Solution of Maze (comparing the backtracking based solution and Dijkstra's algorithm)
5. Different exact and approximation algorithms for Travelling-Sales-Person Problem
6. Creation of Maze using backtracking
7. Knight tour algorithms
8. Network flow optimization and maximum matching
9. AI for different games such as minesweeper, shooting games, Hex, connect-4, sokoban, etc
10. SUDOKU solver
11. Graph theoretic algorithms
12. Computational Geometry Algorithms
13. AKS primality testing
14. Algorithms for factoring large integers
15. Randomized algorithms for primality testing (Miller-Rabin, Solovay-Strassen)
16. Slider puzzle game

List of Course Seminar Topics:

1. Divide and Conquer Vs Dynamic Programming
2. Greedy strategy
3. NP-hardness
4. Backtracking strategy
5. Dynamic Programming Vs Greedy
6. Computational Complexity
7. Philosophical relevance of P Vs NP question
8. Complexity classes
9. Space complexity
10. Compression Techniques
11. Real world applications of Graph theoretic algorithms
12. Approximation algorithms
13. Hardness of approximation
14. Pseudorandom number generators

List of Course Group Discussion Topics:

1. Greedy Algorithms
2. Dynamic Programming strategy
3. Dynamic Programming Vs Greedy
4. NP-completeness
5. P Vs NP question
6. Algorithm design paradigms
7. Different Searching techniques
8. Backtracking strategy
9. Relevance of Cook-Levin theorem
10. Randomness in computation
11. Approximation Algorithms
12. Application of Recursion

List of Home Assignments:**Design:**

1. Problem solving based on Divide and Conquer strategy
2. Problem solving based on Dynamic Programming strategy
3. Problem solving based on Greedy strategy
4. Problem solving based on Backtracking strategy
5. Problems on Randomized Algorithms
6. Problems on Approximation Algorithms
7. Problems on NP completeness

Case Study:

1. AKS primality test
2. Quadratic sieve factoring algorithm
3. Huffman Encoding, LZW encoding
4. Network flow optimization algorithms
5. Approximation algorithms for TSP
6. Cook-Levin theorem and its relationship with intractability of computational problems
7. Sorting techniques

Blog

1. Approximation Algorithms
2. Randomized Algorithms
3. Computational Geometry Algorithms

4. Number Theoretic Algorithms
5. Graph Theoretic Algorithms
6. P Vs NP Problem
7. Complexity classes
8. Greedy Algorithms
9. Divide and Conquer Vs Dynamic Programming

Surveys

1. Primality Testing Algorithms
2. Integer Factoring Algorithms
3. NP-complete problems
4. Compression Techniques
5. Shortest Path Algorithms
6. Algorithms for finding Minimum Weight Spanning Tree
7. SAT solvers

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

Text Books: (As per IEEE format)

1. Cormen, Leiserson, Rivest and Stein “Introduction to Algorithms”, PHI 3rd edition, 2009. ISBN 81-203-2141-
2. Jon Kleinberg, Eva Tardos “Algorithm Design”, Pearson, 1st edition, 2005. ISBN 978-81-317-0310-6
3. Dasgupta, Papadimitriou, Vazirani “Algorithms” McGraw-Hill Education; 1 edition (September 13, 2006), ISBN-10: 9780073523408, ISBN-13: 978-0073523408

Reference Books: (As per IEEE format)

1. Motwani, Raghavan “Randomized Algorithms”, Cambridge University Press; 1 edition (August 25, 1995), ISBN-10: 0521474655, ISBN-13: 978-0521474658
2. Vazirani, “Approximation Algorithms”, Springer (December 8, 2010), ISBN-10: 3642084699, ISBN-13: 978-3642084690

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

The student will be able –

- 1) To formulate computational problems in abstract and mathematically precise manner
- 2) To design efficient algorithms for computational problems using appropriate algorithmic paradigm
- 3) To analyze asymptotic complexity of the algorithm for a complex computational problem using suitable mathematical techniques.
- 4) To establish NP-completeness of some decision problems, grasp the significance of the notion of NP-completeness and its relationship with intractability of the decision problems.
- 5) To understand significance of randomness, approximability in computation and design randomized algorithms for simple computational problems and design efficient approximation algorithms for standard NP-optimization problems.
- 6) To incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solutions for complex computing problems.

CO PO Map:

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2	PO3	PO7	PO12	PSO1
2	3	3	2	1	3

CO attainment levels:

CO No.	1	2	3	4	5	6
--------	---	---	---	---	---	---

Attainment level	1	3	2	3	4	5
-------------------------	----------	----------	----------	----------	----------	----------

Future Courses Mapping:

Following courses can be learned after successful completion of this course:

Advanced Algorithms, Computational Complexity, Computational Geometry, Algorithmic Number Theory, Algorithmic Graph Theory

Job Mapping:

Algorithm design lie at heart of any Computer Science/Engineering application. Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic studying any other domain (in computer science or otherwise) which demands logical thinking. Algorithm design is an essential component of any job based on programming. All Industries in computer Engineering always look for a strong knowledge in Algorithm design and Data structures. If student wants to pursue higher education/ research in Computer Science, this course is must.

Syllabus Template

CS3204::Automated Software Engineering

Course Prerequisites: Data Structures

Course Objectives:

1. To summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices
2. To discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.
3. To formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework
- 4.To compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices using UML-supported modeling tools.
5. To comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics
- 6.To propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance in order to realize system artifacts with the help of Model Driven Development practices using, scheduling, estimation and risk management activities.

Credits:.5.....

Teaching Scheme Theory:...3 Hours/Week

Tut: 1 Hours/Week

Lab:...2.Hours/Week

Course Relevance: Given that software engineering is built upon the foundations of both computer science and engineering, a software engineering curriculum can be approached from either a computer science-first or software engineering-first perspective; there clearly is merit in both approaches. Software engineering spans the entire software lifecycle - it involves creating high-quality, reliable programs in a systematic, controlled, and efficient manner using formal methods for specification, evaluation, analysis and design, implementation, testing and maintenance. any software products are among the most complex of man-made systems, requiring software development techniques and processes that successfully scale to large applications which satisfy timing, size, and security requirements all within acceptable timeframes and budgets. For these reasons, software engineering requires both the analytical and descriptive tools developed in computer science and the rigor that the engineering disciplines

bring to the reliability and trustworthiness of the systems that software developers design and implement while working cohesively in a team environment.

SECTION-1

Topics and Contents

Software Engineering Paradigms: Overview of Software Engineering, Software Process Framework, Traditional Process Models, Process Models: Code-and-Fix, Waterfall Model, Rapid Application Development, Incremental Models, Evolutionary Models, Iterative Development, The Unified Process, Cleanroom Methodology, Component-Based Software Engineering, CMMI, Software Engineering Principles and Practices, Requirements Engineering Tasks, Requirement Elicitation Techniques, Software Requirements: Functional, Non-Functional, Domain Engineering activities, Requirements Characteristics and Characterization, Eliminating Requirement Ambiguities, Conflict Identification and Resolution, Requirement Qualities, Requirement Specification, Requirement Traceability, Requirement Prioritization, Relationship of Requirement Engineering to other Framework Activities, System Scope Determination and Feasibility Study, Statement of Work Generation, Requirements Verification and Validation, Requirement Maturity, Technical Reviews, Stakeholder Management

Overview of Agile Methodology: Introducing Agile in Practice, Landscape of Agile and Planned Methods, Agile Challenges in Practice, Composite Agile Method and Strategy (CAMS), Composite Agile and IT: Enablement, Development, and Maintenance, Collaborative-Agile Business Management, Business Analysis and Composite Agile, CAMS Project Management and ICT Governance, Agile Adoption in Organizations. Time-Boxing, Kanban, and Theory of Constraints, Lean IT, Pair Programming, Extreme Programming, DSDM, User Requirements in the context of Agile

The Scrum: Scrum Origins: What Is Scrum? Scrum Origins, Why Scrum? Scrum Framework, Agile Principles, Overview, Variability and Uncertainty, Sprints., Requirements and User Stories, Product Backlog, Estimation and Velocity, Technical Debt, Roles: Product Owner, ScrumMaster, Development Team, Scrum Team Structures, Managers, Planning: Scrum Planning Principles, Portfolio Planning, Envisioning (Product Planning), Release Planning (Longer-Term Planning), Sprinting: Sprint Planning, Sprint Execution, Sprint Review, Sprint Retrospective, Scrum and Service Industry

SECTION-II

Topics and Contents

System Behavior Specification: Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Building Domain Model, and capturing system behavior in use cases, Use cases and User Stories, Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, Refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization, Communication Diagram, Interaction Overview Diagrams, Timing Diagrams

Software Architecture Design and Configuration Management: Analysis Concepts, Analysis Methods, The Design Model, Design Qualities, Characteristics of Design activities, Design Principles, Cohesion and Coupling, Software Architecture Vs Software Design, Software Reuse, Design Heuristics, User Interface Design: Rules, User Interface Analysis and Steps in Interface Design, Design Evaluation, Source Code Management, Foundations of Software Architecture, Reference Architectures, Architectural Design: Software Architecture, Data Design and Architectural Design, Views, Viewpoints, Perspectives, Conceptual Architecture View, Module Architecture View, Execution Architecture View, Code Architecture View, Architecture styles: Repository, Layered, Pipe-Filter, Call-Return, Peer-Peer, Publish-Subscribe, Client-Server, Two-Tier, Three-Tier, N-Tier, Heterogeneity in Architecture, Categorizing classes: entity, boundary and control, Modeling associations and collections, Preserving referential integrity, Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces,

Project Management Principles and Design Patterns:, Design Patterns: Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, Antipatterns, Applications of Design Patterns, Project Management Activities, Structures and Frameworks, Teamwork, Leadership, Project Planning, Project Scheduling, Risk Analysis, Introduction to Function Points, Empirical Estimation, COCOMO II model.

List of Tutorials: (Any Four)

1. Study of Requirement Engineering
2. Study on preparation of System Requirement Specification
3. Scrum Artifacts
4. User Stories and Use Case
5. Product Backlog Development

6. Burn-up and Burn-down chart development and management
7. Software System Analysis and Design: UML
8. Incorporation of Design patterns

List of Practical's: (Any Eight)

1. A real-world problem issue is required to be identified with manageable scope. The problem scenarios are required to be identified for target system to be developed. The scenarios are stated in the form of Statement-of-Work template. The SOW document shall address the vision, goals, and objectives of the project.
2. The initial requirements and feature set for the target system is required to be identified. The requirements are required to be synthesized with stakeholder participation. The project roles are assigned to the project team with clear indicator of responsibilities. The initial requirements summary document with adequate and minimal infrastructure is required to be developed using multiple iterations.
3. The product backlog for the project aimed at maintaining a prioritized queue of project requirements shall be created.
 - a. It should be dynamic and should be continuously groomed as the project progresses. Agile projects generally use an iceberg strategy for grooming the product backlog.
 - b. The items that are near the top of the iceberg and are closest to going into development should get the most attention.
 - c. There should typically be about two to three sprints worth of stories at the top of the backlog that are well-groomed and ready to go into development in order to avoid a situation where the project team is waiting for work to do.
4. Sprint-level planning activity accommodating story points, planning poker shall be performed. The Sprint-plan and Sprint-design indicating detailed activity planner shall be developed.
5. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior of the target system and map requirements to Use cases.
 - a. The System Context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass
 - b. Actors (External Users)
 - c. Transactions (Use Cases)
 - d. Event responses related to transactions with external agents.
 - e. Detection of System boundaries indicating scope of system.
6. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
 - a. Discrete, distinguishable entities (class).
 - b. Events (Individual stimulus from one object to another).
 - c. Conditional events and relationship representation.
7. To depict the state transition with the life history of objects of a given class model. The model should depict:
 - a. Possible ways the object can respond to events from other objects.

- b. Determine of start, end, and transition states.
8. To depict the dynamic behavior using detailed Activity diagram. Activity is a parameterized behavior represented as coordinated flow of actions. The flow of execution is modeled as activity nodes connected by activity edges.
 - a. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents.
 - b. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions.
9. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
 - a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation
 - b. Identification of objects and their purpose.
 - c. Roles / responsibilities entities that determine system behavior.
10. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.

To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.

List of Projects:

1. Automated Parking lot identifier
2. Health Care Softwires
3. Financial Domain
4. Appraisal Systems
5. Automate Project Administration System
6. Translator for Agriculture System
7. Development of applications manageable by Agile

Development of SMART applications

List of Course Seminar Topics:

1. Agile software development
2. AI and software engineering
3. Apps and app store analysis
4. Automated reasoning techniques
5. Autonomic and (self-)adaptive systems
6. Big data
7. Cloud computing
8. Component-based software engineering

9. Computer-supported cooperative work
10. Configuration management and deployment
11. Crowd sourced software engineering
12. Cyber physical systems
13. Data-driven software engineering
14. Debugging

Dependability, safety, and reliability

List of Course Group Discussion Topics:

1. Distributed and collaborative software engineering
2. Domain modeling and meta-modeling
3. Education
4. Embedded software
5. Emerging domains of software
6. Empirical software engineering
7. End-user software engineering
8. Fault localization
9. Formal methods
10. Green and sustainable technologies
11. Human and social aspects of software engineering
12. Human-computer interaction
13. Knowledge acquisition and management
14. Machine learning for software engineering
15. Middleware, frameworks, and API

List of Home Assignments:

Design:

1. Software visualization
2. Specification and modeling languages
3. Tools and environments
4. Traceability
5. Ubiquitous and pervasive software systems
6. Validation and verification

Case Study:

1. Software economics and metrics
2. Software engineering for machine learning
3. Software evolution and maintenance
4. Software modeling and design
5. Software process
6. Software product lines

Blog

1. Mining software engineering repositories
2. Mobile applications
3. Model-driven engineering

4. Parallel, distributed, and concurrent systems
5. Performance
6. Program analysis
7. Program comprehension
8. Program repair
9. Program synthesis
10. Programming languages
11. Recommendation systems
12. Refactoring

Surveys

1. Requirements engineering
2. Reverse engineering
3. Safety-critical systems
4. Scientific computing
5. Search-based software engineering
6. Security, privacy and trust
7. Software architecture
8. Software reuse
9. Software services
10. Software testing

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

MSE(15)+ESE(15)+HA(10)+LAB(10)+CP(10)+CVV(20)+SEMINAR(10)+GD(10)

Text Books: (As per IEEE format)

1. Ian Sommerville, 'Software Engineering', Addison-Wesley, 9th Edition, 2010, ISBN-13: 978-0137035151.
2. Kenneth S. Rubin, *Essential SCRUM: A Practical Guide To The Most Popular Agile Process*, Addison-Wesley, ISBN-13: 978-0-13-704329-3, 2012
3. Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049

Reference Books: (As per IEEE format)

1. SorenLauesen, *Software requirements: Styles and techniques*, Addison Wesley, ISBN 0201745704, 2002
2. Dean Leffingwell, *Agile Software Requirements*, Addison-Wesley, ISBN-13: 978-0-321-63584-6, 2011
3. Charles G. Cobb, *The Project Manager's Guide To Mastering Agile: Principles and Practices for an Adaptive Approach*, Wiley Publications, ISBN: 978-1-118-99104-6 (paperback), ISBN 978-1-118-99177-0 (epdf), 2015
4. Grady Booch, James Rumbaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974
5. Erich Gamma, Richard Helm, Ralph Johnson, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Professional, ISBN-10: 0201633612 ISBN-13: 978-0201633610
6. Paul Clements, Felix Bachmann, Len Bass, David Garlan, *Documenting Software Architectures: Views and Beyond* Addison-Wesley Professional 2003, ISBN-10:0201703726, ISBN-13: 9780201703726

Moocs Links and additional reading material: www.nptelvideos.in

www.nptelvideos.in

www.coursera.com

www.udemy.com

Course Outcomes:

1. Summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices focusing tailored processes that best fit the technical and market demands of a modern software project.
2. Discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.
3. Formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework along with Propose and demonstrate realistic solutions supported by well-formed documentation with application of agile roles, sprint management, and agile architecture focusing project backlogs and velocity monitoring.
4. Compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices using UML-supported modeling tools.
5. Comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics, published guidance, applicability, reasonableness, and relation to other design criteria resulting in well-documented system profiles to the engineering and social community.
6. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance in order to realize system artifacts with the help of Model Driven Development practices using, scheduling, estimation and risk management activities.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO8	PO11	PSO3
2	3	3	2	1	3

CO attainment levels

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO8	PO11	PSO3
1	5	2	3	3	4

Future Courses Mapping:

Software testing and Quality Assurance, Service-oriented Software

Job Mapping:

Application Architect, Project Designer, SCRUM Role Players

CS3206 :: DATA SCIENCE**Course Prerequisites: Data Structures, Mathematics and Statistics.****Course Objectives:**

- 1 To learn the concepts of descriptive, prescriptive and predictive data analysis.
- 2 To apply quantitative modelling and data analysis techniques to the solution of real world business problems, communicate findings, and effectively present results using data visualization techniques.
- 3 Apply ethical practices in everyday business activities and make well-reasoned ethical business and data management decisions.
- 4 Demonstrate knowledge of statistical data analysis techniques utilized in business decision making.
- 5 Apply principles of Data Science to the analysis of business problems to build machine intelligence.
- 6 Demonstrate use of teamwork, leadership skills, decision making and organization theory.

Credits:..5.....**Teaching Scheme Theory:...3 Hours/Week****Tut: 1 Hours/Week****Lab:...2.Hours/Week**

Course Relevance: The principal purpose of Data Science is to find patterns within data. It uses various statistical techniques to analyze and draw insights from the data. From data extraction, wrangling and pre-processing, a Data Scientist must scrutinize the data thoroughly. Then, he has the responsibility of making predictions from the data. The goal of a Data Scientist is to derive conclusions from the data. Through these conclusions, he is able to assist companies in making smarter business decisions.

SECTION-1

Topics and Contents

Introduction: Data, Big Data, characteristics of Big Data, Data Analytics Lifecycle, structured, unstructured and semi structured data, Data representation, data storage, LaGrange's methods, interpolation, Data Scientist Role, application areas.

Descriptive Statistics: Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Measuring Data Similarity and Dissimilarity, Data Visualization, Prescriptive analysis.

Data Pre-processing: An Overview, Data Cleaning, Data normalization, Data Integration, Data Reduction, Data Transformation and Data Discretization

Predictive Analytics: Linear Regression, Logistic Regression, Association Rules, Optimization Techniques, Hypothesis Testing, confidence interval, Sampling theory and Standard Error.

SECTION-II

Topics and Contents

Classification: Supervised learning, Bayesian Classification, k-Nearest Neighbours (k-NN), Decision tree, Support Vector Machines, Neural Networks, Overfitting/Underfitting, bagging / boosting and ensemble methods, Classifier performance measures, confusion matrix, Cross validation.

Clustering and Outlier Detection: Unsupervised learning, K-means algorithm, hierarchical clustering, Interpretation of clusters and validation, Introduction to outlier detection. Model Planning, Model Building, Decision Support Systems. Chatbots, Recommender systems.

List of Tutorials: (Any Three)

1. Preprocessing of data: consider 10 data samples: Calculate Mean, mode, Median, Standard deviation and Normalized Values.
2. Min Max Normalization: consider 10 data samples: Calculate Normalized Values.
3. Linear Regression: consider 8 trees with different height and trunk diameter. Calculate correlation coefficient. Build a mathematical model and calculate tree height if the new trunk diameter is known.
4. Sampling: Consider iris dataset and apply all types of sampling techniques.
5. Hypothesis testing: apply Null and alternate hypotheses on given statements or problems.
6. Detection and classification: design one best model for Detection and classification of fruits.
7. Clustering: consider 10 persons height and weight and group using K=2 and K=3.
8. Decision Support Systems
9. Real time applications: Recommender system

List of Practical's: (Any Six)

- 1 Getting Started with Python installation and first program
- 2 Data pre-processing and normalization.
- 3 Correlation and Linear regression analysis.
- 4 Data optimizations Using Genetic algorithms.
- 5 Classification of diabetic Data using SVM
- 6 K-means clustering on any one type of UCI Machine learning dataset / Kaggle dataset
- 7 Social media analysis: Twitter/Facebook/WhatsApp: sentiment analysis.
- 8 Building Hadoop Map Reduce Application for counting frequency of words/phrases in simple text file.
- 9 Creating List in python: merging two lists, adding matrices in lists, adding vectors in list.
- 10 Manipulating and Processing Data in Python: merging data sets, sorting data, plotting data, managing data using matrices and data frames

11 Sentiment Analysis of WhatsApp / twitter / facebook data using Python

List of Projects:

- 1 Visit a nearby retail shop, discuss how they are analyzing data, find out limitations and propose solutions for better profit.
- 2 IMDb Movie Analysis
- 3 Creditworthiness of Customers
- 4 Fraud Detection
- 5 Social Media Listening
- 6 Telecom Churn
- 7 Retail-Giant Sales Forecasting
- 8 Interactive Marketing Campaign Analysis
- 9 Inventory management E Commerce
- 10 Stock market price prediction
- 11 Product Delivery Drones
- 12 Smart city water / light management system
- 13 Human Tracking system
- 14 Automatic Interview Conduction system
- 15 Sentiment analysis
- 16 Customer Targeted E-Commerce
- 17 Spam detection
- 18 Recommenders systems

List of Course Seminar Topics:

1. Cyber-crime detection from online social media data
2. Text and product label reading for blind persons
3. Network Intrusion Detection
4. Hate speech recognition
5. Time series data analytics
6. Smart voting system support through face recognition
7. Emotion detection system
8. AWS Data analysis
9. Tsunami Warning system
10. Real time speech recognition
11. Data science future

List of Course Group Discussion Topics:

1. Machine Learning Vs Data science
2. Management information system and Data science
3. Artificial Intelligence and Data science
4. Artificial Intelligence applications
5. Artificial Intelligence future
6. Artificial Intelligence after 10 years / 2030
7. Uninformed searching and Informed searching Techniques
8. Chatbots and Recommender systems
9. Will Automation and Data science Reduce or Increase Jobs.

10. Cashless Economy using Data science
11. Data science in covid-19 situations
12. Image classification vs. video classification
13. Descriptive Prescriptive Analysis
14. Regression and Generalization

List of Home Assignments:

Design:

- 1 DS Accessibility Design
- 2 Best DS for any one real time application
- 3 DS Customer Experience Design
- 4 DS Data-Informed Design
- 5 DS Decision Architecture
- 6 DS Designing for Senior Citizens
- 7 DS in ecommerce Design
- 8 DS Enterprise UX Design
- 9 DS Experience Design
- 10 Sentiment analysis design
- 11 Fitness prediction
- 12 Fake news detection
- 13 Credit card fraud detection
- 14 Road vehicle traffic analysis
- 15 Crime detection

Case Study:

1. ELKAN Medicine
2. How Indian Retail Giant Is Using Data science to Prepare For The 4th Industrial Revolution
3. Rolls-Royce And Google Partner To Create Smarter, Autonomous Ships Based On

Data science.

4. The Amazing Ways Tesla Is Using Big Data and Data science

5. The Incredible Ways John Deere Is Using Data science to Transform Farming

6 .RealDirect (online real estate firm)

7.. Voice quality measurement

8. Home security

9. Tracking and positioning of Human

10. Customer retention

Blog

1. DS Trends

2. DS Research

3. DS sentimental analysis / Chatbot

4. Chatbot Magazine

5. AI Medical / Agriculture

6. NLP applications

7. Energy prospecting

8. Weather/climate change monitoring

9. Entertainment pattern identification

10. Agriculture Industry trends

Surveys

1. Adaption of DS in 2020

2. DS in Industry

3. DS in Digital Marketing
4. DS in Gaming
5. DS after Covid-19
6. DS Agriculture
7. IBM Watson
8. Crowd sourced analytics
9. supply chain analytics and issues
10. Data Science for smart cities
11. Data Science in supply chain management

Suggest an assessment Scheme:

HA, PPT, GD, ,MSE, ESE, LAB, CVV

Text Books: (As per IEEE format)

- 1 Cathy O'Neil and Rachel Schutt. *Doing Data Science, Straight Talk From The Frontline*. O'Reilly. 2014.
- 2 Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition.
- 3 David Dietrich, Barry Hiller, "Data Science & Big Data Analytics", EMC education services, Wiley publications, 2012
- 4 Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning", Springer, Second Edition, 2011.

Reference Books: (As per IEEE format)

- 1 Peng, Roger D and Elizabeth Matsui, "The Art of Data Science." A Guide for Who Works with Data. Skybrude Consulting 200 (2015): 162.
- 2 Evans, James R., and Carl H. Lindner, "Business analytics: the next frontier for decision sciences." Decision Line 43.2 (2012): 4-6.
- 3 James, G., Witten, D., Hastie, T., Tibshirani, R. An introduction to statistical learning with applications in R. Springer, 2013.
- 4 Albright, Winston "Business Analytics: Data Analysis and decision making" Cengage Learning
- 5 Sahil Raj, "Business Analytics", Cengage Learning
- 6 Camm, Cochran, Fry, Ohlmann, Anderson, Sweeney, Williams, "Essentials of Business Analytics" Cengage Learning

Moocs Links and additional reading material:

<https://nptel.ac.in/courses/106/106/106106179/>

<https://nptel.ac.in/courses/110/106/110106064/>

<https://www.coursera.org/specializations/jhu-data-science>

<https://www.coursera.org/professional-certificates/ibm-data-science>

Course Outcomes:

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

- 1 Select appropriate data visualizations to clearly communicate analytic insights to business sponsors and analytic audiences
- 2 Deploy the Data Analytics Lifecycle to address big data analytics projects
- 3 Reframe a business challenge as a Data scientist / developer
- 4 Apply appropriate analytic techniques and tools to analyze big data, create statistical models, and identify insights that can lead to actionable results
- 5 Use tools such as: R and R Studio, MapReduce/Hadoop, in-database analytics,
- 6 Explain how advanced analytics can be handled to create competitive applications.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
-----	-----	-----	-----	-----	-----

PO1	PO3	PO4	PO7	PO11	PSO3
2	3	3	2	1	3

CO attainment levels

CO1-4 CO2-3 CO3-3 CO4-5 CO5-2 CO6-1

Future Courses Mapping:

Data Analytics, Business Analytics, Artificial Neural Networks, Machine Learning, Deep Learning.

Job Mapping:

Data Analyst, Data Scientist, Data Designer, Data Developer

CS3207::Compiler Design

Course Prerequisites: Automata Theory (grammar)

Course Objectives:

1. Understand the process of program execution cycle.
2. Understand the translation process from High Level Languages to Machine Level Language.
3. Know the syntax and semantic analysis approaches for efficient code/program verification.
4. Learn the methods of code generation which helps for the optimization.
5. Learn code optimization and runtime code synthesis.
6. Know the process of compiler design for emerging programming languages.

Credits:4.....

Teaching Scheme Theory:...3 Hours/Week

Tut: Hours/Week

Lab:...2.Hours/Week

Course Relevance: All high level programming languages are easy for users to understand but not understood by a computing machine. The computing machine knows only binary data. A translation is required, in this case, to convert higher level language into machine level, so that the intended program could execute. This translation is done by using a compiler. This course will give you detailed insights of how compilers function internally and design it efficiently. This gives freedom to design your own programming language with it's compiler.

SECTION-1

Topics and Contents

Compilers: Introduction to compiler phases, introduction to cross compiler, features of machine-dependent and independent compilers, overview of types of compilers. Interpreters: compiler vs. interpreter, phases, and working, Preprocessor: header file and macro expansion.

Assembler: Elements of assembly language programming, design of the assembler, assembler design criteria, types of assemblers, two-pass assemblers, one-pass assemblers, assembler algorithms, multi-pass assemblers, variants of assemblers design of two-pass assembler, machine-dependent and machine-independent assembler features.

Linkers: Relocation and linking concepts, static and dynamic linker, subroutine linkages.

Loaders: Introduction to the loader, loader schemes: compile and go, general loader scheme, absolute loaders, relocating loaders, direct linking loaders, MS DOS linker.

Lexical Analysis and introduction to Syntax Analysis: Introduction to Compiler, Phases and Passes, Bootstrapping, Role of a Lexical Analyzer, Specification and Recognition of Tokens, LEX/FLEX, Expressing Syntax, Top-Down Parsing, Predictive Parsers. Implementing Scanners, operator precedence parsers.

Syntax and Semantic Analysis: Bottom-Up Parsing, LR Parsers: constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, YACC/BISON Type Checking, Type Conversion. Symbol Table Structure.

SECTION-II

Topics and Contents

Syntax-Directed Translation and Intermediate Code Generation: Syntax-Directed Definitions, Bottom-Up Evaluation, Top-Down Translation, Intermediate Representations, Intermediate Code Generation. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors, semantic errors. More about translation: Array references in arithmetic expressions, case statements, introduction to SSA form

Code Generation: Issues in Code Generation, Basic Blocks and Flow Graphs, Next-use information, A simple Code generator, DAG representation of Basic Blocks, Peephole Optimization. Generating code from dags.

Code Optimization and Run-Time Environments: Introduction, Principle Sources of Optimization, Optimization of basic Blocks, Introduction to Global Data Flow Analysis, Runtime Environments, Source Language issues. Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter Passing

Machine Dependant Optimization: Instruction (Basic-Block) scheduling algorithm, Instruction selection algorithm, Register allocation techniques, peephole optimizations

Introduction to Data flow analysis: Introduction to constant propagation, live range analysis

Case studies: LLVM compiler Infrastructure, Power of SSA, compiling OOP features, Compiling in multicore environment, Deep learning compilation,

List of Practical's: (Any Six)

- 1) LEX/FLEX specification and programming regular expressions
- 2) Add line numbers to lines of text, printing the new text to the standard output using LEX/FLEX.
- 3) Implement LEX/FLEX code to select only lines that begin or end with the letter 'a' and delete everything else.
- 4) Implement LEX/FLEX code to count the number of characters, words and lines in an input file.
- 5) Implement LR/SLR/LALR Parser.
- 6) Implement Syntax directed Translator.
- 5) Convert all uppercase characters to lowercase except inside comments.
- 6) Change all numbers from decimal to hexadecimal notation, printing a summary statistic (number of replacements) to stderr.
- 7) Implement Lexical Analyzer for language C-.
- 8) YACC specifications and implement Parser for specified grammar.
- 9) Implement Parser for language C-.
- 10) Implement an Intermediate code generator (three address code and Quadruples)

List of Projects:

1. Compiler for subset of C using Lex and YACC
2. Compiler for Subset of Java programming Language
3. Intermediate Code generator
4. Code Optimizer
5. Develop an Editor for Assembly programming. (Use available Assembler MASM/TASM to compile the code and execute in editor)
6. Design a system to check syntax and semantics of English Language.
7. Design a system to check syntax and semantics of a subset of Logical programming Language.
8. Design a System to check syntax and semantics of a subset of Python programming language.
9. Compiler for subset of C++ programming language
10. Compiler for a subset of Algol programming language

List of Course Seminar Topics:

1. Tools complementary to Lex
2. Tools complementary to YACC
3. Semantic Analyser
4. Obsolete programming Language compiler advantage and issues
5. Android App program compiler
6. Approaches of Intermediate Code generation
7. Recent Trends in Compiler
8. Recent Trends in Interpreter
9. Decompilation
10. Compilation in multicore machines

List of Course Group Discussion Topics:

1. Compiler Vs Interpreter
2. Multi Language Compiler
3. Tree structure for parsing
4. Decompilers: Good or Bad
5. Universal Compiler
6. Cross compiler
7. Alternate to parsers
8. Compiler challenges in mobile app development.
9. Online Compilers

10. Compilers in field of Game development

List of Home Assignments:

Design:

1. Recent methodologies in Intermediate Code Generator
2. Recent methodologies in Code Optimizer
3. Universal Compiler
4. Compiler for Deep learning
5. Recent trend in parsers

Case Study:

1. Algol Compiler
2. Compilation process(internals) of Functional Programming
3. Compilers for Mobile App development
4. LLVM compiler
5. Cross compiler

Blog

1. Decompilers: Ethical or Unethical?
2. Multiparadigm programming compiler
3. State of the Art tools for rapid compiler development
4. Compiler for parallel machines
5. Compiler for distributed computing

Surveys

1. Obsolete Programming Language Compilers
2. Obsolete Programming Language Interpreter
3. Compilers for various programming paradigms
4. Online compilers
5. Mobile app cross compiler

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360-degree assessment and check if it covers all aspects of Bloom's Taxonomy.

MSE(15)+ESE(15)+HA(10)+LAB(10)+CP(10)+CVV(20)+SEMINAR(10)+GD(10)

Text Books: (As per IEEE format)

1. Aho, A.V., Lam, M.S., Sethi, R., & Ullman, J.D. (2006). *Compilers: Principles, Techniques, and Tools*, Addison Wesley, ISBN 978-81317-2101-8 (2nd Edition).
2. Cooper, K., & Torczon, L. (2011). *Engineering a compiler*. Morgan Kaufmann, ISBN 155860-698-X.
3. Appel, A. W. (2004). *Modern compiler implementation in C*. Cambridge university press.
4. Appel, A. W., & Jens, P. (2002). *Modern compiler implementation in Java*. In ISBN 0-521-58388-8. Cambridge University Press.
5. Appel, A. W. (1998). *Modern Compiler Implementation in ML*, In ISBN 0-521-60764-7. Cambridge University Press.
6. Raghavan, V. (2010). *Principles of Compiler Design*. Tata McGraw-Hill Education.

Reference Books: *(As per IEEE format)*

1. Muchnick, S. (1997). *Advanced compiler design implementation*. Morgan Kaufmann, ISBN 8178672413
2. Levine, J. R., Mason, J., Levine, J. R., Mason, T., Brown, D., Levine, J. R., & Levine, P. (1992). *Lex & yacc*. "O'Reilly Media, Inc".

Moocs Links and additional reading material: www.nptelvideos.in

https://swayam.gov.in/nd1_noc20_cs13/preview

<https://www.udacity.com/course/compilers-theory-and-practice--ud168>

<https://online.stanford.edu/courses/soe-yccscs1-compilers>

Course Outcomes:

- 1) Design basic components of a compiler including scanner, parser, and code generator.
- 2) Perform semantic analysis in a syntax-directed fashion using attributed definitions.
- 3) Apply local and global code optimization techniques.
- 4) Synthesize machine code for the runtime environment.
- 5) Develop software solutions for the problems related to compiler construction.
- 6) Adapt themselves to the emerging trends in language processing.

CO PO Map

CO1-PO2 - 2

CO2-PO3 - 3

CO3-PO4 - 3

CO5-PO11 - 2

CO6-PO12 - 1

CO4-PSO3 - 3

CO attainment levels

CO1 – 2, CO2 – 3, CO3 – 3, CO5 – 3, CO4 – 4, CO6 - 5

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

Software Engineer, Compiler Developer

Syllabus Template**CS3211:: Engineering Design-I****Course Prerequisites: Problem Based Learning****Course Objectives:**

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Credits:.1.....**Teaching Scheme Theory:... Hours/Week****Tut: 2 Hours/Week****Lab:....Hours/Week**

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guidelines mentioned herewith. For all courses of ED, laboratory course contents of “Engineering Design” are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone à Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Topic and Content: Programming Robots and Drones Using C++ and Python

The MODBUS protocol, an industry standard, is used frequently throughout the robot industry. Robot-to-Robot communication method. ROS is an open-source, meta-operating system for your robot. Control logic for robots using infrared and ultrasonic sensors, Proximity sensors and light sensors. Communication architectures and protocols for networking unmanned aerial vehicles (UAV). Protocols - Common Data Link (CDL), Tactical Common Data Link (TCDL), Link-11, Link-14, Link-16, and Link-22, Micro Air Vehicle Communication Protocol (MAVLink) used for bidirectional communications between the drone and the GCS.

List of Practical's:

Lab can be conducted in real and simulation mode.

Programming a simulated robot is analogous to programming a real robot.

- 1) Design layout with proximity and define all the states to simulate autonomous mobile robots.
- 2) Deploy the IR sensors in layout and achieve the communication among them
- 3) Deploy the Ultrasonic in layout and achieve the communication among them
- 4) Deploy other required sensors and achieve the communication among them
- 5) Design layout with proximity and define all the states to simulate autonomous mobile drone

(Unmanned Aerial Vehicles-UAV).

6) Deploy the base station and achieve the communication with the sensors of UAV using point to point MAVLink protocol

7) Deploy the base station and achieve the communication with the sensors of UAV using point to multi point link protocol

...not limited to.....**Faculty and students are free to include other area which meets the society requirements at large.**

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Focus on the higher levels of the Booms Taxonomy analyze, apply, evaluate and create.

Text Books: (As per IEEE format)

1. Name(s) of author(s); Title of the book; Edition No., Publisher

2

3

4

Reference Books: (As per IEEE format)

1. Name(s) of author(s); Title of the book; Edition No., Publisher

2

3

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO4	PO5	PO7	PO12	PSO1
2	3	3	2	1	3

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

What are the Job opportunities that one can get after learning this course

Syllabus Template

CS3210::Engineering Design and Innovation-III**Course Prerequisites: Problem Based Learning****Course Objectives:**

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Credits:.4.....**Teaching Scheme Theory:... Hours/Week****Tut: Hours/Week****Lab:...Hours/Week**

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

SECTION-1

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of “Engineering Design” are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards à Single Board Computers and IoT à Computational Biology (Biomedical and Bioinformatics)à Robotics and Drone à Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.
- It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:

Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.

This is the sample list to start with. Faculty and students are free to include other area which meets the society requirements at large.

The model begins with the identifying of a problem, often growing out of a question or “wondering”. This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.

- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students’ wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an

interdisciplinary approach in both the analysis and solving phases.

- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have the ability to initiate the task/idea .They should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?

- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use following mechanism to maintain the track of moving towards the solution.
- How effective is? How strong is the evidence for? How clear is?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDI Sample Case Studies : -

Lab can be conducted in real and simulation mode. Programming a simulated robot is analogous to programming a real robot.

- 1) Design and deployment of autonomous mobile robot in either real mode or simulation mode.
- 2) Design and deployment of unmanned aerial vehicles-UAV in either real mode or simulation using point to point protocol.
- 3) Design and deployment of unmanned aerial vehicles-UAV in either real mode or simulation using point to multipoint protocol.

...not limited to....Faculty and students are free to include other area which meets the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

Higher levels of the Booms Taxonomy - analyze, apply, evaluate and create.

Text Books: (As per IEEE format)

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. De Graaff E, Kolmos A., red.: *Management of change: Implementation of problem-based and project-based learning in engineering*. Rotterdam: Sense Publishers. 2007.
2. *Project management core textbook, second edition, Indian Edition*, by Gopalan.
3. *The Art of Agile Development*. By James Shore & Shane Warden.

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO4	PO5	PO7	PO12	PSO1
2	3	3	2	1	3

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

What are the Job opportunities that one can get after learning this course