College of Engineering, Pune

(An Autonomous Institute of Govt. of Maharashtra, Permanently Affiliated to S.P. Pune University)

Department of Computer Engineering and I.T.

Curriculum Structure & Detailed Syllabus (UG Program)

Final Year B. Tech. Computer Engineering

(Effective from: A.Y. 2022-23)

Sr. No.	Item	Page No
1	Programme Education Objectives (PEOs) , Programme Outcomes (POs) and Programme Specific Outcomes (PSOs)	2
2	Correlation between PEOs, PSOs and POs	3
3	List of Abbreviations	3
4	Curriculum Structure, List of: Electives, Minor, Honor courses	4
5	Table of Contents	8
6	Detailed Syllabi	10-

B.Tech. Computer Engineering

Programme Educational Objectives (PEOs):

I. To create graduates with sufficient capabilities in computer engineering who can become researchers, entrepreneurs and software professionals to satisfy the needs of the core industry, research, academia and society at large.

II. To build ability to continuously learn the latest trends in computer engineering and engage in lifelong learning process.

III. To build engineers aware of professional ethics of the software Industry, and equipped with basic soft skills essential for working in community and professional teams.

Programme Outcomes (POs):

At the end of the program, the graduates will

1. **Computer engineering knowledge:** Apply the knowledge of mathematics, science, computer engineering fundamentals, and emerging fields of computer engineering to the solution of complex real-life problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex computer engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and computer engineering sciences.
3. **Design/development of solutions:** Design solutions for complex computer engineering problems and design system components or processes that meet the specified needs considering public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern computer engineering and IT tools including FOSS tools.
6. **Social responsibility:** Apply reasoning informed by the contextual knowledge to assess social, health, safety, legal and cultural issues and the consequent responsibilities.
7. **Environment and sustainability:** Understand the impact of the professional computer engineering solutions in socio-environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Demonstrate knowledge and practice of engineering ethics.
9. **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary, multi-cultural settings.
10. **Communication:** Communicate effectively with engineering community and with society at large, demonstrating ability to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the computer engineering, finance and management principles.
12. **Life-long learning:** Recognize the need for, and ability to engage in independent and life-long learning

Programme Specific Outcomes (PSOs)

Students will be able to

1. Demonstrate competence in Programming Technologies.
2. Design, implement, test software solutions in core Computer Engineering areas including Computer Networks, Databases, Systems Software, Computer Architecture, Artificial Intelligence, Software Engineering
3. Acquire and demonstrate skills in emerging area like Information Security, Data Science, Natural Language Processing, Cloud Computing, etc.

Correlation between the PEOs and the POs

PO→ PEO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
I	▪	▪	▪	▪	▪					▪	▪	▪
II	▪	▪	▪	▪	▪	▪	▪	▪	▪	▪	▪	▪
III						▪	▪	▪	▪	▪	▪	

Correlation between the PEOs and the PSOs

PSO→ PEO↓	I	II	III
I		▪	▪
II	▪		▪
III			

List of Abbreviations

Sr. No.	Abbreviation	Stands for:
1	BSC	Basic Science Course
2	DEC	Departmental Elective Course
3	HSMC	Humanities, Social Sciences and Management Course
4	IFC	Interdisciplinary Foundation Course
5	IOC	Interdisciplinary Open Course
6	LC	Laboratory Course
7	MLC	Mandatory Learning Course
8	PCC	Programme Core Course
9	SBC	Skill Based Course

SCHEME-A (CE)

Semester VII : Scheme A

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1	MLC	ML-22001	Intellectual Property Rights	1	0	0	0
2	LLC	LL-2200x	Liberal Learning Course	1	0	0	1
3	IOC	IOC-2200x	Interdisciplinary Open Course-II (**Management related courses** to be offered by Humanities/Production Engineering Department)	2	0	0	2
4	PCC	CT-22001	Compiler Construction	3	0	0	3
5	LC	CT-22005	Compiler Construction Laboratory	0	0	2	1
6	PCC	CT-22003	Cryptography and Network Security	3	0	0	3
7	LC	CT-22004	Cryptography and Network Security Laboratory	0	0	2	1
8	DEC	CT(DE)-220xx	Department Elective-II	3	0	0	3
9	DEC	CT(DE)-220xx	Department Elective-III	3	0	0	3
10	LC	CT(DE)-220xx	Department Elective-III Laboratory	0	0	2	1
			Total	**16**	**0**	**6**	**18**
			Total Academic Engagement and Credits	22			

Semester VIII : Scheme A

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1	SLC	CE(DE)-220xx	Department Elective-IV	3	0	0	3
2	SLC	CE(DE)-220xx	Department Elective-IV Laboratory	0	0	2	1
3	SBC	CT-22006	In-house Project	0	0	16	8
			Total	**3**	**0**	**18**	**12**
			Total Academic Engagement and Credits	21			

Minor / Honours courses: As mentioned on separate page

Of the four Department Elective courses, the student shall be taken at least two courses in MOOCs mode

SCHEME-B (CE)

Semester VII : Scheme B

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1	MLC	ML-22001	Intellectual Property Rights	1	0	0	0
2	LLC	LL-2200x	Liberal Learning Course	1	0	0	1
3	IOC	IOC-2200x	Interdisciplinary Open Course-II (**Management related courses** to be offered by Humanities/Production Engineering Department)	2	0	0	2
4	PCC	CT-22001	Compiler Construction	3	0	0	3
5	LC	CT-22005	Compiler Construction Laboratory	0	0	2	1
6	PCC	CT-22003	Cryptography and Network Security	3	0	0	3
7	LC	CT-22004	Cryptography and Network Security Laboratory	0	0	2	1
8	DEC	CT(DE)-220xx	Department Elective-II	3	0	0	3
9	DEC	CT(DE)-220xx	Department Elective-III	3	0	0	3
10	LC	CT(DE)-220xx	Department Elective-III Laboratory	0	0	2	1
			Total	**16**	**0**	**6**	**18**
			Total Academic Engagement and Credits	22			

Semester VIII : Scheme B

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1	SLC	CT(OC)-22001	Massive Open Online Course -I	2	0	0	2
2	SLC	CT(OC)-22002	Massive Open Online Course -II	2	0	0	2
3	SBC	CT-22006	Major Project with Industry/Corporate/Academia/Open-Source Communities	0	0	16	8
			Total	**4**	**0**	**16**	**12**
			Total Academic Engagement and Credits	20			

Minor / Honours courses: As mentioned on separate page

Of the four Department Elective courses, the student shall be taken at least two courses in MOOCs mode

List of Department Electives

Department Elective-II

Course code	Course Title
CT(DE)-22001	FP: Functional Programming
CT(DE)-22002	CBD: Cloud and Big Data
CT(DE)-22003	IR: Information Retrieval
CT(DE)-22004	SDP: Software Design Patterns
CT(DE)-22005	DS: Distributed Systems
CT(DE)-22006	OOMD: Object Oriented Modelling and Design
CT(DE)-22007	ADBMS: Advanced Database Management Systems
CT(DE)-22008	WT: Wireless Technologies
-	Subjects in Association with Domain Experts

Department Elective-III

Theory Course Code	Laboratory Course Code	Course Title
CT(DE)-22009	CT(DE)-22018	IOT: Internet of Things
CT(DE)-22010	CT(DE)-22019	Devops
CT(DE)-22011	CT(DE)-22020	DFDR: Digital Forensics and Data Recovery
CT(DE)-22012	CT(DE)-22022	STQA: Software Testing and Quality Assurance
CT(DE)-22013	CT(DE)-22022	AUP: Advanced UNIX Programming
CT(DE)-22014	CT(DE)-22024	ES: Embedded Systems
CT(DE)-22015	CT(DE)-22023	NLP: Natural Language Processing
CT(DE)-22016	CT(DE)-22025	PCAP: Parallel Computer Architecture and Programming
CT(DE)-22017	CT(DE)-22026	CS: Cyber Security
-	-	Subjects in Association with Domain Experts

Department Elective-IV

Theory Course Code	Laboratory Course Code	Course Title
CT(DE)-22027	CT(DE)-22033	SA: System Administration
CT(DE)-22028	CT(DE)-22034	S&V: Storage and Virtualization
CT(DE)-22029	CT(DE)-22035	MAN: Mobile and Ad-hoc Networks
CT(DE)-22030	CT(DE)-22036	CB: Computational Biology
CT(DE)-22031	CT(DE)-22037	MT: Multicore Technology
CT(DE)-22032	CT(DE)-22038	GIS: Geographical Information Systems
CT(DE)-22039	CT(DE)-22041	Introduction of Blockchains, Cryptocurrencies, and Smart Contracts
CT(DE)-22040	CT(DE)-22042	GPU Computing
-	-	Subjects in Association with Domain Experts

Minor in Computer Engineering

(To be offered to students of other departments)

SN	Semester	Course Code	Course Name	Lectures-Tutorial-Lab-Credits
1	V	CT(MI)-21001	Data Structures, Files and Algorithms	3-0-0-3
2	VI	CT(MI)-21002	Object Oriented Programming and Design	3-0-0-3
3	VII	CT(MI)-22001	Internet Technologies	3-0-0-3
4	VIII	CT(MI)-22002	Data Science (Minor)	3-0-0-3

Honours in Data Science

SN	Semester	Course Code	Course Name	Lectures-Tutorial-Lab-Credits
1	V	CT(HO)-21001	Making Sense of Data	3-0-0-3
2	VI	CT(HO)-21003	Big Data Analytics	3-0-0-3
3	VII	CT(HO)-22001	Deep Learning	3-0-0-3
4	VIII	CT(HO)-22003	Reinforcement Learning	3-0-0-3

Honours in Information Security

SN	Semester	Course Code	Course Name	Lectures-Tutorial-Lab-Credits
1	V	CT(HO)-21002	Fundamentals of Information and Coding Theory	3-0-0-3
2	VI	CT(HO)-21004	Ethical Hacking	3-0-0-3
3	VII	CT(HO)-22002	Malware Analysis	3-0-0-3
4	VIII	CT(HO)-22004	IOT Security	3-0-0-3

Table of Contents

Core Courses 10

(CT-22001) Compiler Construction 10

(CT-22005) Compiler Construction Laboratory 11

(CT-22003) Cryptography and Network Security 12

(CT-22004) Cryptography and Network Security Laboratory 13

Departmental Elective – II 14

(CT(DE)-22002) Cloud Computing and Big Data 14

(CT(DE)-22001) Functional Programming 17

(CT(DE)-22003) Information Retrieval 19

(CT(DE)-22004) Software Design Patterns 21

(CT(DE)-22006) Object Oriented Modeling and Design 23

(CT(DE)-22005) Distributed Systems 25

(CT(DE)-22007) Advanced Database Management Systems 26

(CT(DE)-22008) Wireless Technologies 28

Departmental Elective – III 29

(CT(DE)-22009) Internet of Things 29

(CT(DE)-22018) Internet of Things Laboratory 31

(CT(DE)-22010) DevOps 33

(CT(DE)-22019) DevOps Laboratory 34

(CT(DE)-22011) Digital Forensics and Data Recovery 35

(CT(DE)-22020) Digital Forensics and Data Recovery Laboratory 36

(CT(DE)-22012) Software Testing and Quality Assurance 37

(CT(DE)-22021) Software Testing and Quality Assurance Laboratory 39

(CT(DE)-22013) Advanced Unix Programming 40

(CT(DE)-22022) Advanced Unix Programming Laboratory 42

(CT(DE)-22014) Embedded Systems 44

(CT(DE)-22024) Embedded Systems Laboratory 46

(CT(DE)-22015) Natural Language Processing 46

(CT(DE)-22023) Natural Language Processing Laboratory 48

(CT(DE)-22016) Parallel Computer Architecture and Programming 49

(CT(DE)-22025) Parallel Computer Architecture and Programming Laboratory 50

(CT(DE)-22017) Cyber Security 51
(CT(DE)-22026) Cyber Security Laboratory 52
Departmental Elective – IV 53
(CT(DE)-22027) System Administration 53
(CT(DE)-22033) System Administration - Laboratory 55
(CT(DE)-22028) Storage & Virtualization 56
(CT(DE)-22034) Storage & Virtualization - Laboratory 58
(CT(DE)-22029) Mobile and Ad-Hoc Networks 58
(CT(DE)-22035) Mobile and Ad-Hoc Networks Laboratory 60
(CT(DE)-22030) Computational Biology 61
(CT(DE)-22036) Computational Biology Laboratory 62
(CT(DE)-22031) Multicore Technology 63
(CT(DE)-22037) Multicore Technology Laboratory 64
(CT(DE)-22032) Geographical Information Systems 65
(CT(DE)-22038) Geographical Information Systems Laboratory 67
(CT(DE)-22039) Introduction to Blockchain, Cryptocurrencies, and Smart Contracts 68
(CT(DE)-22041) Introduction of Blockchain, Cryptocurrencies, and Smart Contracts Laboratory 69
(CT(DE)-22040) GPU Computing 70
(CT(DE)-22042) GPU Computing Laboratory 71
MINOR in Computer Engineering 72
(CT(MI)-22001) Internet Technologies (Minor) 72
(CT(MI)-22002) Data Science (Minor) 73
HONOURS in Computer Engineering 75
(CT(HO)-22001) Deep Learning (Honours) 75
(CT(HO)-22003) Reinforcement Learning (Honours) 76
HONOURS in Information Security 78
(CT(HO)-22002) Malware Analysis 78
(CT(HO)-22004) Internet of Things Security 79

Core Courses

(CT-22001) Compiler Construction

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 Hrs/week	Assignment/Quizzes – 40 marks End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the understanding of different phases of compilation.
2. Demonstrate the ability to generate and code lexical and syntax analyzer.
3. Analyze and differentiate different parsing techniques and syntax directed translation schemes and choose the optimal parsing technique.
4. Apply different intermediate code generation representation for program construct.
5. Identify different code optimization techniques for the various statements

Introduction: The role of language translation in the programming process, Translator Issues, Types of translators, Design of an assembler Introduction to phases of compilation, Front end and Back end model of compiler, Cross compiler, Incremental compiler.

[6 Hrs]

Lexical Analysis: Concept of Lexical Analysis, Regular Expressions, Tokens, Lexemes, and Patterns, Block Diagram of Lexical analyser, Revision of finite automata, Introduction to LEX Tool and LEX file specification, Error detection and recovery in LEX.

[6 Hrs]

Syntax Analysis: Context Free Grammars(CFG),Concept of parsing, Parsing Techniques, Top-Down Parsers : Introduction, Predictive Parsing - Removal of left recursion, Removal of left factoring, Recursive Descent Parsing, Predictive LL(k) Parsing Using Tables, Bottom Up parsing: Introduction, Shift-Reduce Parsing Using the ACTION/GOTO Tables, Table Construction, SLR(1), LR(1), and LALR(1) Grammars, Introduction to YACC Tool & YACC file specification, Error detection and recovery in YACC.

[8 Hrs]

Semantic Analysis & Intermediate Representation: Need of semantic analysis, Abstract Parse trees, Syntax directed definitions, Syntax directed translation schemes, Symbol Tables, Symbol Table management, Data type as set of values and with set of operations Type Checking, Intermediate code generation: Intermediate languages, Design issues, Intermediate code representations: three address code, abstract & concrete syntax trees,.

[8 Hrs]

Code Optimization: Introduction, Principal sources of optimization, Machine Independent

Optimization, Machine dependent Optimization, Various Optimizations: Function preserving transformation, Common Sub-expressions, Copy propagation, Dead-code elimination, Loop Optimizations, Code Motion, Induction variables and strength reduction, Peephole Optimization, Redundant –instruction elimination.

[6 Hrs]

Run-Time Memory Management & Code generation: Model of a program in execution, Stack and static allocation, Activation records , Issues in the design of code generation, Target machine description, Basic blocks & flow graphs, Expression Trees, Unified algorithms for instruction selection and code generation.

[6Hrs]

Text Books:

- Alfred V. Aho, Monica S. Lam, A. V. R. Sethi and J.D. Ullman, "Compiler Principle, Techniques and Tools" Addison Wesley, Second Edition, ISBN: 978-0321486813.

Reference Books:

- Barrent W. A., J. D. Couch, "Compiler Construction: Theory and Practice", Computer Science series, Asian student edition, ISBN: 9780574217653.
- Dhamdhere D.M., "Compiler Construction Principle and Practice", Macmillan India, New Delhi, 2003, Second Edition, ISBN: 0333904060.
- Ravendra Singh, Vivek Sharma, Manish Varshney, "Design and Implementation of Compiler", New Age Publications, 2008, ISBN: 978-81-224-2398-3.
- Holub, A.J., "Compiler design in C", Prentice Hall, 1994, ISBN: 978-0133049572.
- John Levine, Tony Mason & Doug Brown, "Lex and Yacc", O" Reilly, 1995, Second Edition, ISBN: 978-1565920002.

(CT-22005) Compiler Construction Laboratory

Teaching Scheme:	**Examination Scheme:**
Laboratory: 2 Hrs/ Week	Continuous evaluation: 50 Marks
	End Sem Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Implement lexical analyzers using Lex tool
2. Write a parser and semantic analyzer for different Context-Free Grammars using Yacc tool.
3. Implement different representations of Intermediate code
4. Demonstrate ability to optimize intermediate code using different techniques

Suggested List of Assignments:

1. Design a lexical analyzer for a subset of C language using Lex tool.
2. Design a hand-coded lexical analyzer for a subset of C language, draw the transition diagrams and then implement the lexical analyzer in C language.
3. Design a scientific calculator using Lex & Yacc or PLY or ANTLR tools.
4. Write a code for finding FIRST & FOLLOW of a grammar.
5. Design a SQL parser / html parser.
6. Implement a SLR parser for a given grammar.
7. Implement a static semantics analyzer.
8. Implement an intermediate code generator in three-address code form represented in quadruples.
9. Implement different optimization techniques on intermediate code.

(CT-22003) Cryptography and Network Security

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Explain the concepts related to applied cryptography, including plaintext, ciphertext, symmetric cryptography, asymmetric cryptography, and digital signatures
2. Apply concepts of finite mathematics and number theory.
3. Demonstrate the understanding of common network vulnerabilities and attacks, defence mechanisms against network attacks, and cryptographic protection mechanisms.
4. Detect possible threats to different defence mechanisms and different ways to protect against these threats

Course Contents

Introduction: Cryptography and modern cryptography, Need of security, Security services, Basic network security terminology, Security attacks, Classical cryptosystems and their cryptanalysis, Operational model of network security

[4 Hrs]

Mathematical Foundations: Prime Number, relatively prime numbers, Modular Arithmetic, Fermat's and Euler's Theorem, The Euclidean and Extended Euclidean Algorithms, The Chinese Remainder Theorem, Discrete logarithms

[6 Hrs]

Symmetric Key Ciphers: Symmetric Key Ciphers, Feistel Networks, Modern Block Ciphers, Modes of Operation, Cryptanalysis of Symmetric Key Ciphers: Linear Cryptanalysis, Differential Cryptanalysis

[8 Hrs]

Asymmetric Cryptography: RSA, Key Distribution and Management, Diffie-Hellman Key Exchange, Elliptic Curve Cryptography, hash functions: The Merkle Damgard Construction, Message Digest algorithms: MD5, Secure Hash algorithm (SHA), Message Authentication Codes

[8 Hrs]

Authentication and Web Security: Digital Signatures, Authentication Protocols, Kerberos, X.509 Digital Certificate Standard, Pretty Good Privacy, Secure Socket Layer, Secure Electronic Transaction. Zero knowledge proof

[8 Hrs]

Network Security: Intruders, Intrusion Detection, Password Management, Worms, viruses, Trojans, Virus Countermeasures, Vulnerabilities in TCP/IP model, Firewalls, Firewall Design Principles.

[6 Hrs]

Text Books

- "Cryptography and Information Security", V. K. Pachghare, 3rd edition, PHI Learning, ISBN: 978-93-89-347-10-4.
- "Network Security: Private Communication in a Public World", Charlie Kaufman, Radia Perlman, and Mike Speciner, Prentice Hall, ISBN 0-13-046019-2.

Reference Books

- "Cryptography and Network Security, Principles and Practices", William Stallings, Pearson Education, Fifth Edition, and ISBN: 0-13-60970-9.
- "Network Security the Complete Reference", Robert Bragge, Mark Rhodes, Heith Straggberg Tata McGraw Hill Publication, ISBN: 9780072226973.

(CT-22004) Cryptography and Network Security Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs/ Week	Continuous evaluation: 50 Marks
	Mini Project: 25 marks
	End Semester Exam: 25 Marks

Course Outcomes

Students will be able to:

1. Analyze the optimal features and time required for an encryption technique.
2. Implement cryptographic algorithms in any programming language.
3. Demonstrate the ability to detect attacks on a system and tackle it.
4. Write a security application to protect a system from some attacks.

Suggested List of Assignments

1. 1. Study papers on a network security topic and write a study report
   a) Wireless Network Security,
   b) Key Exchange Protocols,
   c) Block chain.
2. 2. Implement any one classical encryption technique in any programming language.
3. 3. Design and implement a symmetric encryption algorithm based on Feistel structure.
4. 4. Demonstrate how Diffie-Hellman key exchange works with Man-In-The-Middle attack.
5. 5. Study different approaches for Anti-virus software and write one document.
   a) Examine files to look for viruses by means of a virus dictionary
   b) Identifying the suspicious behavior from any computer program which might indicate infection
6. 6. Study and demonstrate system hacking and write a report.
   a) How to crack a password?
   b) How to use Ophcrack / Crowbar / John the Ripper / Aircrack-ng to Crack Passwords
7. 7. Develop a mini project on
   a) a hack tool to break the security of a system.
   OR
   b) a tool to protect the system from the hack tool.

This is a suggested list. The instructor is expected to continuously update it.

Departmental Elective – II

(CT(DE)-22002) Cloud Computing and Big Data

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Comprehend basic concepts of cloud computing and virtualization.
2. Identify various cloud-based solutions to meet a set of given requirements.
3. Visualize development of applications using kubernetes and container concepts.
4. Gain fundamentals of big data and big data processing frameworks.
5. Demonstrate applications of Apache framework for big data processing and analysis on cloud.

Course Contents

Introduction: History of Centralized and Distributed Computing - Overview of Distributed Computing, Cluster computing, Grid computing Distributed Computing and Enabling Technologies, Cloud Fundamentals: Cloud Definition, Evolution, Architecture, Applications, deployment models, and service models, case-study of cloud computing systems like Amazon EC2 and S3, Google App Engine, and Microsoft Azure etc.

[6 Hrs]

Virtualization: Virtualization basics, virtualization technologies and architectures, types of virtualization, para-virtualization, full-virtualization, Internals of virtual machine monitors/hypervisors, Issues with virtualization and Issues with Multi-tenancy, Introduction to Micro-services, Containers and Kubernetes, Micro services architecture, benefits of micro services, characteristics of micro services, migration to micro services, CoreOS and OpenShift.

[8 Hrs]

Migration and Interoperability: Broad Aspects of Migration into Cloud, Migration of virtual Machines and techniques, Fault Tolerance Mechanisms, Interoperability approaches, Service Monitoring, Issues with interoperability, Vendor lock-in, SLA Management, Metering Issues.

[6 Hrs]

Resource Management and Load Balancing: Server consolidation, Dynamic provisioning and resource management, Resource Optimization, Resource dynamic reconfiguration, Scheduling Techniques, Capacity Management to meet SLA Requirements, and Load Balancing, load balancing techniques.

[7 Hrs]

Cloud Orchestration: Introduction to Cloud Orchestration, backing up VM, restoring VM, clustering and high availability, VM migration, Case-study of OpenStack: NOVA, Neutron, Keystone Cinder, Swift and Glances, CI / CD.

[8 Hrs]

Big Data: Big Data for cloud, Characteristics of Big data, Key aspects of a Big Data Platform, Apache Hadoop, MapReduce - HDFS framework ,Spark Ecosystem Walkthrough, Overview of HBase, HBase Data-Model, Overview of Apache Hive and Apache Pig.

[5 Hrs]

Text Books

- "Virtualization, A beginners Guide", Danielle Ruest and Nelson Ruest, Tata McGraw Hill, ISBN: 978-661-2031-892
- "Moving to the cloud", Dinakar Sitaram and Geetha Manjunath, Elsevier, ISBN: 978-159-7497-251
- "Using Docker: Developing and Deploying Software with Containers",Adrian Mouat, O'reilly Media, ISBN: 978-149-1915-769.
- "Hadoop: The Definitive Guide", Tom White, O'reilly Media, ISBN: 978-144-9311-520.

Reference Books :

- "Cloud Computing: Methodology, Systems, and Applications", L. Wang, R. Ranjan, J. Chen, and B. Benatallah, CRC Press, Boca Raton, FL,USA, ISBN: 978-143-9856-413

- "Cloud Computing: Principles and Paradigms", Buyya R., Broberg J., Goscinski A., John Wiley & Sons Inc., ISBN: 978-0-470-88799-8
- "Cloud Computing",V.K. Pachghare, PHI Publication, ISBN: 978-81-203-5213-1
- "The Docker Book: Containerization is the new virtualization", James Turnbull, ISBN: 978-098-8820-203
- "Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results", Bernard Marr, Wiley, ISBN: 978-111-9231-387

Online Resources:

- http://docs.openstack.org/
- http://mininet.org/
- https://www.opennetworking.org/
- http://pubs.vmware.com/vsphere-50/index.jsp
- https://developers.google.com/appengine/
- http://www.windowsazure.com/en-us/
- http://www.cloudfoundry.com/
- http://aws.amazon.com/developers/getting-started/

Suggested List of Assignments

1. To create and run virtual machines on open source OS [VirtualBox, VMWare Workstation].
   a) To install an operating system in the virtual machine from template
   b) Add storage to create the new virtual disk.
2. To install OpenStack and use it as Infrastructure as a Service.
   a) Create and delete compute resources.
   b) Attach volumes to running instances
   c) Create a network and subnet for the web server nodes.
3. To install hypervisor such as KVM, ESXi.
   a) Deploy VM on hypervisor
   b) Back up or migrate VM.
4. To create AWS EC2 Instances.
   a) Logging into the AWS portal
   b) To attach and detach an EBS volume to an EC2 instance.
   c) To create an S3 Bucket for object storage to EC2 instance.
5. To configure and start VM from azure portal.
   a) Logging into the Microsoft Azure portal
   b) Connect to the Windows virtual machine using Remote Desktop Protocol (RDP) software.
6. Deploy a stateless / stateful application on kubernetes cluster.
7. Perform setting up and Installing Hadoop in its three operating modes:
   a) Standalone Mode
   b) Pseudo Distributed Mode
   c) Write Program using Hadoop Spark for Word Count Example, Weather Data Set Example.

    d) Write a Pig Latin script to handle the Weather Data Set problem.[sort, group, join, project, and filter Weather data]

8. To Create Hive tables and Querying Hive tables for Weather Data Set problem.
    a) Create, alter, and drop databases, tables, views, functions, and indexes.
9. To Create HBase tables that include multiple column families and columns
    a) Alter properties associated with a table
    b) Populate a table with data
    c) Retrieve data from a table
10. To create and query HBase tables using BigSQl
    a) Populate these tables using Big SQL INSERT and LOAD operations
    b) Query these tables using projections and restrictions.

This is an illustrative list of assignments. The instructor is expected to update the list.

(CT(DE)-22001) Functional Programming

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Identify difference between various programming paradigms
2. Application of lambda calculus and combinatory logic in programming
3. Evaluate functions written in functional programming language
4. Analyze memory management in functional programming language
5. Application of functional programming in problem solving

Course Contents

Introduction to Programming Paradigms. Programming languages, Design Issues of Programming Language, Programming Paradigms, Necessity, Types, Ideas behind imperative, applicative, object oriented and logic programming.

[8 Hrs]

Haskell illustrating the following concepts. Functions as first class objects, referential Transparency, Laziness , Data types and pattern matching, Types, Classes and overloading, Side effect free IO

[6 Hrs]

Functional Programming Paradigm. First class and higher order functions , Combinatory logic, Intuitionistic logic, Lambda Calculus : Syntax of Lambda terms Alpha, beta and delta conversions Normal forms, applicative order and normal order reductions Church-Rosser theorems Y combinator and recursion.

[6 Hrs]

Functions in functional programming. Inbuilt functions, user defined functions, high order functions on list, recursive function, lazy function evaluation, polymorphism, File I/O, Channels and Concurrency, pattern matching

[6 Hrs]

Type inference. Hindley-Milner type checking algorithm and its extension to handle type classes.
Implementation of functional programming. Issues in implementation, the spineless-tagless G-Machine,

[8 Hrs]

Other topics. Bird-Meertens Formalism, Abstract Interpretation, Functional Programming using Monads.
Application of functional programming. Hadoop, MapReduce, Google, Facebook , APIs in other langages like PHP, Java, Ruby.

[8 Hrs]

Text Books

- "Programming in Haskell", Graham Hutton, Cambridge University Press, ISBN, 978-0521692694.
- "Implementing functional languages: A Tutorial" , Simon Peyton Jones and David R. Lester, Publisher: University of Glasgow, ASIN: B001UHUR8W

Reference Books

- "Programming Languages: Concepts and Constructs", Ravi Sethi, Pearson Education; 2 edition, ISBN 978-8177584226.
- "Programming Languages: Design and Implementation", T W Pratt ,Marvin V. Zelkowitz , Pearson; 4th edition, ISBN 978-0130276780
- "Performance and Evaluation of Lisp Systems", The MIT Press; ISBN 0-262-07093-6.
- "Haskell vs. Ada vs. C++ vs. Awk vs. ... An experiment in Software Prototyping Productivity" , Paul Hudak and Mark. P. Jones, Technical report, Department of Computer Science, Yale University, New Haven, CT 06518.
- "Why Functional Programming Matters?", John Hughes, Technical Memo of Institutionen for Datavetenskap, Chalmers Tekniska Hogskola, Goteborg, Sweden.
- 'Haskell : the craft of functional programming", *Thompson, Simon,* 3rd edition, Harlow: Addison-Wesley, 2011
- 'Learn you a Haskell for great good! : a beginner's guide", *Lipovača, Miran,* San Francisco, CA: No Starch Press, cop. 2011

Suggested List of Assignments

1. Implement a program in functional language to demonstrate various data structures
2. Implement a program in functional languages to create, read elements traverse elements and delete elements from a list.

3. Write a function to find factorial of a number in functional programming languages.
4. Write a program which uses in-built functions in functional programming language.
5. Write a program to read and write file.
6. Write a program to demonstrate concurrency in functional programming.
7. Write a program to demonstrate polymorphism in functional programming.
8. Write a program to demonstrate classes in functional programming.

Some more assignments can be added by the instructor. The Haskell language is recommended for programming.

(CT(DE)-22003) Information Retrieval

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Recognize basic concepts and techniques in Information Retrieval.
2. Design data structures such as an index, to allow efficient access to the information in large bodies of text.
3. Analyze different approaches for query processing.
4. Analyze how statistical models of text can be used to solve problems in IR, with a focus on how the vector-space model and language models are implemented and applied to document retrieval problems.
5. Implement a relevance feedback mechanism to improve system performance.

Course Contents

Introduction: Motivation, Data retrieval versus information retrieval, Goals and history of IR, Problems in IR, Modeling- taxonomy of IR models, Retrieval- adhoc and filtering, Basic Models of IR Boolean, Vector-space model.

[7 Hrs]

Basic preprocessing and Indexing: Simple tokenizing, Stop-word removal, Stemming, Inverted index and postings, Text-similarity metrics, TF, IDF, TF-IDF (term frequency/inverse document frequency) weighting, k-Shingles, Distance and similarity measures- Minkowski, Manhattan, Euclidean, Jaccard, Pearson, Cosine similarity measure.

[6 Hrs]

Query Operations: Relevance feedback- Global methods and local methods, Global methods - Query expansion/reformulation with a thesaurus or WordNet, Query expansion via automatic thesaurus generation, Techniques like spelling correction, Local methods - Relevance feedback, Pseudo relevance feedback, Indirect relevance feedback, Rocchio algorithm for relevance feedback, Probabilistic relevance feedback, Kullback-Leiber divergence (KL-divergence) retrieval function.

[8 Hrs]

Indexing and Searching: Introduction, Inverted Files - Construction, Searching, Suffix trees and suffix arrays, Signature files, Boolean queries, Sequential searching- Brute Force, Knuth Morris-Pratt, Shift-or, Suffix Automation, Pattern matching- Regular expression and extended patterns, Structural queries, Compression- sequential searching, compressed indexing

[8 Hrs]

Probabilistic Retrieval and Evaluation: Probabilistic Model, Formal model for enhancing retrieval effectiveness by using sample information about the frequency of occurrence and co-occurrence of index terms in the relevant and non-relevant documents, Traditional view of measurement of effectiveness, Performance metrics: recall, precision, and F-measure, Evaluations on benchmark text collections.

[6 Hrs]

Link Analysis: The Web as a graph, PageRank, Topic specific PageRank, Hubs and Authorities.

[5 Hrs]

Text Books

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze, "Introduction to Information Retireval", Online Edition, Cambridge University Press, ISBN: 978- 0521865715.
- Richardo Baeza –Yates, Berthier Ribiero-Neto "Modern Information Retrieval " Addison Wesley, Second Edition, ISBN: 978-0321416919.
- C J Van Rijsbergen, "Information Retrieval", An online book by C J Van Rijsbergen, University of Glasgow, ISBN: 978-0408709293.

Reference Books:

- Kowalski, Gerald J, "Information Retrieval Systems Theory and Implementations", Springer, ISBN: 978-0-585-32090-8
- David A. Grossman, Ophir Frieder, "Information Retrieval Algorithms and Heuristics", Springer, ISBN: 978-81-8128-917-9
- Jin Zhang, "Visualization for Information Retrieval", Springer, ISBN-13: 978-3540751472.

Suggested List of Assignments

1. Design and Implement pre-processing module required for building an inverted index file.
2. Building an Inverted Index File for the data on machine.
3. Design and Implement the query processing module of an search engine.
4. Design and implement a web-based crawler.
5. Design and implement different distance and similarity measures (Euclidean, Jaccard, Manhattan, Cosine) required for finding duplicates or near-duplicates in the datasets.
6. Installation and study of elastic Search Engine.

1. Design Search Engine using Solr and Lucene for your college website.

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22004) Software Design Patterns

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Create software designs that are scalable and easily maintainable to understand the use of Object Oriented concepts.
2. Apply creational design patterns in software design for class instantiation and devise structural design patterns for object composition.
3. Apply behavioral patterns for organization and communication between the objects
4. Propose refactoring to compose the methods for proper code packaging and to organize the class responsibilities of current code

Course Contents

Introduction: What is a design Pattern? Design patterns in smalltalk MVC, Describing Design patterns, the catalog of design patterns, organizing the catalog, How design patterns solve design problems, how to select a design pattern, how to use a design pattern.

[6 Hrs]

Designing a document editor: Case Study on Design Problems, Document Structure, formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, supporting multiple window system, user operations, spelling checking and hyphenation

[6 Hrs]

Creational Patterns: Abstract Factory, builder, factory Method, prototype, Singleton.

[8 Hrs]

Structural Patterns: Adapter, bridge, composite, decorator, façade, flyweight, proxy.

[8 Hrs]

Behavioral Patterns: Chain of Responsibility, command, interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

[6 Hrs]

Patterns and Software Architecture: Architectural pattern, Difference between design patterns and architectural patterns, different architectural patterns, relation between software architecture and patterns.

[6 Hrs]

Text Books

- "Design Patterns: Elements of Reusable Object Oriented Software", Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Pearson 1st edition. ISBN-13: 078-5342633610

- “Design Patterns Explained- A New Perspective on Object Oriented Design”, Allan Shalloway, James Trott, Addison Wesley 2nd Edition, ISBN-13: 978-0321247148

Reference Books

- “Head First Design Patterns”, Eric Freeman and Elisabeth Freeman, O'Reilly, 2004. ISBN-13. 978-0596007126
- “Applied Java Patterns”, Stephen Stelting and Olav Maassen, Prentice Hall, 2002. ISBN, 0130935387, 9780130935380
- “Java Design Patterns - A Tutorial”, James W. Cooper, Addison-Wesley, 2000. ISBN : 0-201-48539-7
  “Refactoring To Patterns”, Joshua Kerievsky, Addison-Wesley, 2005. ISBN, 0321213351, 9780321213358

Suggested List of Assignments

Implementation of programs can be done using programming languages like Java, C++.

The design representation can be done using any UML tool. Students can choose system of their choice for implementation / design purpose.

1. Design the organization of catalog along using the following design patterns.
   a) Creational Patterns.
   b) Structural Patterns.
   c) Behavioral Patterns.
2. Write a program to implement the following concepts in java.
   a) Method overriding.
   b) Interface.
   c) Abstract class.
3. For Library management system (system of student’s choice)
   a) Write a Program to implement Factory pattern.
   b) Write a program to implement abstract factory.
4. 4. For Banking system
   a) Write a Program to implement Singleton pattern.
   b) Write a Program to implement Composite design pattern
5. 5. For online examination system
   a) Write a Program to implement decorator pattern..
   b) Write a program to implement proxy design pattern.
6. 6. For Stock prediction system
   a) Write a Program to design chain of responsibility pattern.
   b) Write a program to design mediator pattern
7. 7. For Railway reservation system
   a) Write a Program to implement iterator pattern.
   b) Write a program to implement visitor pattern.
8. 8. User gives a print command from a word document.

a) Design to represent this chain of responsibility
b) Design flyweight pattern, façade pattern.

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22006) Object Oriented Modeling and Design

Teaching Scheme:
Lectures: 3 Hrs/Week

Examination Scheme:
Assignments/Quizzes: 40 Mark
End Term Exam: 60 Marks

Course outcomes

Students will be able to:

1. Specify, analyze and design the use case driven requirements for a particular system.
2. Identify, formulate and solve software development problems: software requirements,
6. specification (problem space), software design, and implementation (solution space).
3. Model the event driven state of object and transform them into implementation specific
7. layouts.
4. Analyze and explore different analysis and design models, such OO Models, Structured Analysis and Design Models, etc.
5. Apply knowledge software engineering methods, such as object-oriented analysis and design methods with a clear emphasis on UML.

Introduction & Modeling Concepts: Rational unified process emphasizing inception, Elaboration, Construction, Transition phases, Architectural approaches, Use case centric, Architecture driven, Iterative approach, Introduction to UML, UML history, UML new features, UML meta model, Extensibility mechanisms like stereotypes, Tagged values, Constraints and Profiles, OCL, Overview of all diagrams in UML.

[8 Hrs]

Advanced Class Modeling: Introduction to object diagrams, Class diagrams, Classes and relationships, Interfaces and ports, Templates, Active objects, Advanced relationships generalization, Inheritance, Association, aggregation, and Dependencies.

[7 Hrs]

Structure Modeling: Composite structure diagrams including composite structures, Collaborations, Interaction diagrams, Interaction overview diagrams including interactions, Signals, Exceptions, Regions, partitions, Sequence diagrams, Communication diagrams.

[7 Hrs]

State Modeling: State machine diagrams, States, Encapsulation of states, Transitions, Submachine, State generalization, Timing diagrams, Activity diagrams, Activities, Sub activities, Signals, Exceptions, Partitions, and Regions.

[7 Hrs]

Interaction Modeling: Support for modelling, Architecture in UML, Package diagrams, Component diagrams, Deployment diagrams, Applications of UML in embedded systems, Web applications, Commercial applications, UML 2.0 for each diagram the Need, Purpose, Concepts, Notation, Forward Engineering.

[7 Hrs]

Databases Modeling: Concepts of distributed operating system, COM and CORBA, Introduction to Object Oriented Database.

[4 Hrs]

Text Books

- Grady Booch, James Rumbaugh, Ivar Jacobson "Unified Modeling Language User Guide", The (2nd Edition) (Addison-Wesley Object Technology Series) (Hardcover), 2013.
- Michael Bilaha, James R Rambaugh: Object Oriented Modelling and Design, PHI, 2007, 2edition, ISBN-10: 8131711064

Reference Books

- Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third Edition (Paperback) ,Addision Wesley, 2011
- Dan Pilone, Neil Pitman "UML 2.0 in a Nutshell", (In a Nutshell (O'Reilly)) Paperback)
- Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado "UML 2 Toolkit (Paperback) "
- Jim Arlow, Ila Neustadt "UML 2 and the Unified Process : Practical Object-Oriented Analysis and
- Design" (2nd Edition) (Addison-Wesley Object Technology Series) (Paperback)

Suggested List of Assignments:

1. Elicit requirements towards the identified project statement.
2. Capture Functional Requirements with Use Cases for the project / problem statement
3. Design Use Case Scenarios and use case diagrams and documentation
4. Perform Static Analysis Modeling and Dynamic Analysis Modeling
5. Perform Static Design Modeling and Dynamic Design Modeling
6. Apply different Design Patterns, Object Constraint Language, Persistence for your problem statement.
7. Draw all UML diagrams for your BTech Project

Students shall submit minimum five UML based implementations on: three models to describe a system from different viewpoints.

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22005) Distributed Systems

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Identify challenges faced while designing a distributed system,
2. Apply appropriate alogrithms in distributed systems in various scenarios,
3. Analyse the trends in distributed systems

Course Contents

Introduction Motivation, goals, advantages , disadvantages, hardware concepts, software concepts, design issues, middleware, overview of distributed systems .

[6 Hrs]

Communication. Client server model, middleware and client server model, relation of Network models with distributed system (TCP/IP, OSI, ATM etc.) , Remote Procedure Call, group communication and its protocol (IS-IS)

[6 Hrs]

Synchronization. Clock synchronization, logical clocks, lamport's algorithm, global state, vector algorithm,election algorithms, mutual exclusion alogrithms, dead locks in distributed systems, deadlock avoidance, prevention and detection.

[8 Hrs]

Models in DS. Threads, system models, processor allocation, workstation model, Processor Pool Model, Hybrid Model, real time distributed systems, time triggered systems, event driven systems, distributed shared memory, consistency models, page based distributed shared memory.

[8 Hrs]

Distributed File System. design, implemetation, trends.

[6 Hrs]

Applications of distributed systems. distributed object based systems, distributed co-ordination based systems, distributed document based systems.

[6 Hrs]

Text Books

- ”Distributed Systems –Principles and Paradigms" , Andrew S. Tanenbaum & Maarten van Steen, PHI, ISBN-978-81-203-2215
- "Distributed Operating Systems Concepts and Design" , Pradeep K. Sinha, PHI Publication

Reference Books

- “*Distributed Operating Systems*", Andrew S. Tanenbaum, Pearson Education

- “Distributed Systems, Concepts and Design” , George Coulouris, Jean Dollimore & Tim Kindberg, Fifth Edition, Pearson.
- “Distributed Operating Systems: Concepts and Practice”, Galli D.L., Prentice-Hall.
- “Distributed Systems”, Mullender S., Addison Wesley.

Suggested List of Assignments

1. Implement a client server model to read/write a file present at single server from multiple clients. Trace packets using tool like tcpdump and show the data which is passed on and from server from and to client.
2. Implement a RPC program to read/write a file present at single server from multiple clients. Trace packets using tool like tcpdump and show the data which is passed on and from server from and to client. Also show the protocol used.
3. Implement election algorithm in distributed system.
4. Implement deadlock detection algorithm for distributed system.
5. Implement synchronization algorithm like lamports algorithm or its variants for clock synchronization in distributed system.
6. Case Study of real time distributed system.
7. Configure of NFS client and server in order to read and write a file present on NFS server. Trace packets using tool like tcpdump and show the data which is passed on and from server from and to client. Also show the protocol used.
8. Implement an application using distributed system techniques like DCOM/CORBA/TIB-RENDEVOUS etc. [mini-project]

This is an illustrative list of assignments. The instructor is expected to update the list.

(CT(DE)-22007) Advanced Database Management Systems

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Examine different parallelism techniques for parallel databases .
2. Analyze Distributed Transaction and Query Processing.
3. Think and analyze parallel processing of queries.
4. Analyze the cost of different SQL joins.
5. Demonstrate NoSQL.

Course Contents

Introduction: Database System Architectures: Centralized and Client-Server Architectures, Three–tier Architecture , Server System Architecture, Parallel System, Distributed System, Network Types.

[6 Hrs]

Parallel Databases : I/O Parallelism, Inter-query and Intra-query Parallelism, Interoperation and Intra-operational Parallelism, Design of Parallel systems.

[6 Hrs]

Distributed Databases: Homogeneous and Heterogeneous databases, Distributed DBMS architectures, Storing data in distributed DBMS, Distributed catalog management.

[6 Hrs]

Distributed Transactions: Distributed Transactions, Distributed Query processing: Non-join queries and joins, Updating distributed data: Synchronous and asynchronous replication, Distributed concurrency and recovery.

[8 Hrs]

Data Warehouse and OLAP: Introduction to Decision support, Data Warehousing, Creating and maintaining a warehouse, OLAP: Multidimensional data model, OLAP Queries, Database design for OLAP, Implementation techniques for OLAP: Bitmap Indexes, Join Indexes, Views and decision support, Top N queries, Online aggregation , View Materialization : Issues.

[8 Hrs]

NoSQL Databases: Searching: Features of NoSQL databases, Types of No SQL databases: Key-Value databases, Columnar Databases~~,~~ Introduction to data Mining.

[6 Hrs]

Text Books

- "Database system concepts", Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 5th Edition, McGraw Hill International Edition, ISBN-10: 9332901384, ISBN-13: 978-9332901384

- "Database Management Systems", Raghu Ramkrishnan, Johannes Gehrke, Third Edition, McGraw Hill India Edition 2014, ISBN-10: 93-392-1311-4, ISBN-13: 978-93-392-1311-4

Reference Books

- "Principles of Database Systems", J. D. Ullman, 2nd Edition, Galgotia Publication, ISBN10: 0716780690, ISBN-13: 978-0716780694 .
- "Fundamentals of Database System", R. Elmasri and S. Navathe, , Pearson, 7th Edition, ISBN-10: 933258270X, ISBN-13: 978-9332582705 .

Suggested List of Assignments

1. Write a program to demonstrate fragmentation and replication of databases.
2. Demonstrate relational operation sort-merge join algorithm.
3. Demonstrate OLAP queries.

4. Design and implement an application to demonstrate distributed databases..
5. Design and Implement a small application for Android Tablet.
6. Hadoop installation and programming using MapReduce.

(CT(DE)-22008) Wireless Technologies

Teaching Scheme	Examination Scheme
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Examine the fundamental concepts of wireless Networking and wireless Communication.
2. Investigate the operation of Radio Frequency communications and their functionality in wireless communications.
3. Investigate the features and installation of wireless personal area networks and local area networks.
4. Analyze wireless metropolitan area network and wide area network.

Course Contents

Introduction to Wireless Networks: IEEE Standards for Wireless Networks. Wireless Networks Applications. Types of Wireless Networks. Benefits of Wireless Networks.
Wireless System Architecture: Wireless System Components, Network Architecture. Information Signals.

[8 Hrs]

Radio Frequency and Light Signal Fundamentals: Wireless Transceivers, understanding RF Signals, Working of Light Signals. Types of Wireless Networks: WPAN, WLAN, WMAN Wireless PAN: Components: User Devices, Radio NIC, USB Adapters, Wireless Routers, Bluetooth Dongles etc.

[6 Hrs]

Wireless PAN Systems: Printing, Accessing Internet, Accessing PDA's, Mobile Phones. Wireless PAN Technologies: IEEE 802.15. Bluetooth Version 1 and Version 2.

[6 Hrs]

Wireless LAN: Meaning, Components: User Devices, Radio NIC's, Access Points, Routers, Repeaters, And Antennae. Wireless MAN: Meaning and Components: Bridges, Bridges Vs. Access Points, Ethernet to Wireless Bridges, Workgroup Bridges.

[6 Hrs]

Wireless MAN Systems: Point to Point Systems, Point to Multi-Point, Packet Radio Systems
Wireless WAN: WAN User Devices, Base Stations, Antennae. Wireless WAN Systems: Cellular-Based Wireless WANs, First-Generation Cellular, Second-Generation Cellular, Third-Generation Cellular.

[6 Hrs]

Space-Based Wireless WAN: Satellites, Meteor Burst Communications. Wireless Networks Security: Security Threats, Unauthorized Access, Middle Attacks, DoS Attack (Denial of Service). Authentication: 802.11 Authentication Vulnerabilities, MAC Filters, Authentication Using Public Key Cryptography, 802.1x , Security Policies.

[8 Hrs]

Text Books

- Garg, "Matthew Gast, "802.11 Wireless Networks" Second Edition, The Definitive Guide, ISBN: 0-596-10052-3
- Vijay Wireless Communications & Networking" Elsevier, ISBN: 978-0-12-373580-5

Reference Books

- John Ross, "Introduction to wireless networks" Second Edition, No Starch Press, ISBN 9781593271695
- Theodore S. Rappaport, "Wireless Communications: Principles and Practice" Second Edition, Pearson, ISBN: 978-81-317-3186-4
- Wolfgang Osterhage, "Wireless Network Security" Second Edition, Kindle Edition, ISBN-13: 978-1138093799
- Clint Smith, Daniel Collins, "Wireless Networks: Design and Integration for LTE, EVDO, HSPA, and WiMAX" 3rd Edition, Kindle Edition, ISBN-13: 978-0071819831

Suggested List of Assignments

1. Characterization of Wireless Channels (Simulation/Experiment)
2. Equalization Techniques for Wireless Channels
3. Simulation / Implementation of Multicarrier Modulation
4. Simulation / Implementation of Space Time Block Codes
5. Performance Studies of Adaptive Modulation and Coding
6. Performance Studies of Random MAC Protocols
7. Performance Studies of LLC Protocols
8. Wireless Routing Protocols
9. IOS development of applications using prototype router boards, switches.
10. Wired and Wireless Packet Analysis using Open Source Tools.
11. Network Security Protocols
12. QoS Analysis on Wireless Networks

This is a suggested list. The instructor is expected to continuously update it.

Departmental Elective – III

(CT(DE)-22009) Internet of Things

Teaching Scheme	**Examination Scheme**
Lectures : 3 Hrs/week	Assignment/Quizzes – 40 marks End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the application areas of IOT
2. Analyze data and knowledge management and use of devices in IoT technology
3. Analyze gateways and data management in IoT.
4. Analyze real world IoT design constraints, Industrial automation and commercial building automation in IoT.

Introduction and concepts: Origins, Drivers, Applications, Physical and logical design of IoT, IoT enabling technologies: Wireless sensor networks, Cloud computing, Big data analytics, Communication protocols, embedded systems, IoT levels and deployment templates.

[8 Hrs]

IoT and M2M: Introduction to M2M, Difference between IoT and M2M, M2M and IoT technology fundamentals: Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a service (XaaS), M2M and IoT analytics, Knowledge management

[7 Hrs]

IoT Architecture-state of the art: Architecture reference model: Introduction, Reference model and architecture, IoT Reference model, M2M to IoT-an architectural overview: Building architecture, Main design principles and needed capabilities, State of the art, Standards considerations.

[7 Hrs]

Data management in IoT: Managing M2M data, Data collection and analysis (DCA), Big data, Semantic sensor networks and semantic annotation of data, Virtual sensors, Complex event processing.

[6 Hrs]

Security, Privacy & Trust: IoT security challenge, Spectrum of security considerations, Unique security challenges of IoT devices, Internet of things privacy background, Unique privacy aspects of internet of things, Trust for IoT.

[6 Hrs]

Developing IoT solutions: Introduction to Python, Introduction to different IoT tools, Introduction to Arduino and Raspberry Pi Implementation of IoT with Arduino and Raspberry, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Case studies in IoT design.

[6 Hrs]

Text Books

- Vijay Madisetti, Arshdeep Bahga, "Internet of Things: A Hands-On Approach", Universities Press (India) Private Limited, 2016, ISBN: 978 81 7371 954 7
- Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis Karnouskos, Stefan Avesand & David Boyle "From Machine-to-Machine to the Internet of Things", Academic Press, Elsevier, 2014, ISBN: 978-0-12-407684-6

Reference Books

- Karen Rose, Scott Eldridge, Lyman Chapin, "The Internet of Things: An Overview", Internet Society, 2015
- Adrian McEwen, Hakim Cassimally, "Designing the Internet of Things", Wiley, 2014, ISBN 978-1-118-43062-0
- Daniel Kellmereit, "The Silent Intelligence: The Internet of Things", 2013, ISBN 0989973700

(CT(DE)-22018) Internet of Things Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs/ Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Demonstrate to program on embedded systems
2. Demonstrate how to communicate with other mobile devices using various communication platforms
3. Analyze estimation of the performance of the developed scalable cloud storage and execution platform for IoT on the real sensors data

IOT Kit contains following items:

1. Arduino Uno Rev 3 (Made in Italy) with USB Cable
2. ESP8266 (ESP01) Serial WiFi Module
3. 1-Channel Relay Module
4. Breadboard (Regular)
5. Assorted Jumper Wires (20)
6. LM35 Temperature Sensor
7. Assorted LEDs (10)

List of Assignments:

1. **Embedded Programming**
   a) Toggling LEDs

b) Transmitting a string through UART
c) Controlling LEDs blinking pattern through UART
d) Echo each character typed on serial terminal.
e) Digital IO configuration.
f) Timer based LEDToggle.
g) On-chip Temperature measurement through ADC.

2. **RF experiments**
   a) Point-to-Point communication of two Motes over the radio frequency.
   b) Multi-point to single point communication of Motes over the radio frequency.

3. **Experiments on interfacing with UbiSense**
   a) I2C protocol study
   b) Reading Temperature and Relative Humidity value from the sensor.
   c) Reading Light intensity value from light sensor.
   d) Reading of atmospheric pressure value from pressure sensor.
   e) Proximity detection with IR LED.
   f) Generation of alarm through BuzzerTransmitting the measured physical value from the UbiSense over the Air.

4. **Experiments on interfacing with Ubi-DAQ**
   a) Timestamp with RTC
   b) IO Expander
   c) Relay control.
   d) I2C based 12-channel ADC
   e) EEPROM read and write

5. **WSN Applications**
   a) Demonstration of a Peer-to-Peer network topology using Coordinator and end device network device types.
   b) Demonstration of Peer-to-Peer communication between Coordinator and end device through Router.
   c) Establishing Many-to-One Communication (Star Network Topology)
   d) Establishing Tree Network Topology
   e) Establishing Cluster Tree Network

6. **IOT applications**
   a) Porting 6LoWPAN stack on Ubimote for enabling it with IPv6
   b) 6LoWPAN network formation with motes and PC
   c) IP based lighting control through Data Acquisition Card
   d) IP based sensor monitoring through Ubi-Sense

Each section any two assignments should be implemented by using IoT research lab kit as per Lab instructor.

(CT(DE)-22010) DevOps

Teaching Scheme
Lectures: 3 Hrs / Week

Examination Scheme
Assignment/Quizzes : 40 marks
End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Describe the steps involved in application lifecycle
2. Discuss the benefits of automating application lifecycle
3. Demonstrate the ability to use tools involved in automating each stage of application lifecycle automation
4. Discuss differences between monolithic and micro-services architecture
5. Implement a micro-service based application using DevOps tools

Course Contents

Introduction: History of DevOps; Definition; Main Objectives; Software Development Life Cycle; Agile Model; **Basics of DevOps Components:** Continuous Integration & Deployment; Jenkins; Containers and Virtual Development; Docker; Vagrant; Configuration Management Tools; Ansible; Puppet; Chef; Kubernetes;

[4 Hrs]

Cloud Computing: Evolution of Cloud Computing; IAAS (Infrastructure as a Service); SAAS (Software as a Service); PAAS (Platform as a Service); Private, Public and Hybrid Cloud; Public Clouds; Amazon Web Services; Microsoft Azure; Google Cloud Services;

[3 Hrs]

Shell, GIT: Shell Scripting: Introduction; Basics of Using GIT

[3 Hrs]

Continuous Integration: Introduction to Jenkins; CI with Jenkins; Configuring Jenkins; Scheduling build Jobs; POLL SCM; Periodical builds; Maven Build Scripts; Support for the GIT; Types Jenkins Jobs; Build Pipe Line; Parent and Child Builds; Sequential Builds; Master & Slave Node Configuration; Workspace Management; Security; Authentication; Authorization; Confidentiality; Creating Users; Plugins; Installing Jenkins Plugins; SCM plugin; Build and test;

[6 Hrs]

Configuration Management: Ansible, Puppet, Shef; Introduction to Ansible; Ansible Server Configuration; Infrastructure Management; SSH Connection; YAML Scripts; Inventory; Hosts and Groups and Variables; Group Variables; Host and Group Specific Data; Ad-hoc Commands; Playbooks; Variables; Conditionals; Handlers; Templates; Modules; Core Modules; Extra Modules; Roles; **Terraform:**Terraform Loops, Built-in Functions,Provisioners; Site Reliability and Scalability

[10 Hrs]

Docker: Images, Containers, Docker Engine, Creating containers from images, Command line, Compose, Docker Hub, Trusted Registry, Swarm, Docker file; Orchestration; Vagrant; Networking; seurity;

[6 Hrs]

Kubernetes: Understanding cluster architecture; Installing the Kubernetes dashboard; Rolling updates; Kubernetes Overview; Setup Kubernetes; Kubernetes Concepts; Kubernetes Concepts - PODs, ReplicaSets; Deployments; Networking in Kubernetes; Services; Microservices Architecture. Edge Computing: Basics of edge computing, Kubedge;

[8 Hrs]

Text Books

- The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations; Author(s): Gene Kim, Jez Humble, Patrick Debois, John Willis; Publisher: IT Revolution Press, Year: 2016; ISBN: 1942788002,9781942788003
- Cloud Native DevOps with Kubernetes, 2nd Edition; Author(s): Justin Domingus; John Arundel; Publisher: O'Reilly Media, Inc., Year: 2022; ISBN: 9781098116804,9781098116828,9;781098116767
- Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale; Author(s): Jennifer Davis, Katherine Daniels; Publisher: O'Reilly Media, Year: 2016; ISBN: 1491926309,9781491926307

(CT(DE)-22019) DevOps Laboratory

Teaching Scheme
Laboratory: 2 Hrs / Week

Examination Scheme
Continuous evaluation: 50 Marks
End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Demonstrate the ability to carry out CI/CD tasks
2. Demonstrate the ability to setup Docker Containers
3. Orchestrate using Kubernetes

Suggested List of Assignments

1. Demonstrate the ability to use git using github/gitlab, and application of push, pull, merge, rebase.
2. Setup a website using PAAS machine provided by some cloud provider.
3. Create a customized Docker image using provided instructions.
4. Demonstrate volume and networking setup using multiple Docker instances
5. Create a kubernetes replicaset involving nginx and Ubuntu Docker images, specifying given RAM and CPU limits, and using a clusterIP for accessing the services

6. Write a Jenkins pipeline to demonstrate automatic compilation and build of a project hosted on gitlab/github.
7. Write a terraform script to create virtual machines with the specified RAM, CPU limits and specified image.
8. Write an Ansible playbook to create specified users, with specified privileges, install specified software on specified list of machines.

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22011) Digital Forensics and Data Recovery

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Identify legal issues of performing digital forensics based on investigator duty/position.
2. Discuss various digital forensics techniques.
3. Analyze different file system used in different operating system.
4. Apply various tools during real world forensics investigation.
5. Aware of the state of the practice and gaps in the technology, policy and legal issues.

Course Contents

Issues in Digital Forensics: Computer Crime and Law, Legal Aspects of Digital Forensics, Forensic Investigation Procedure, Investigation Techniques, Digital Forensic Evidence, Anti-forensics, Computer Forensic Model, Maintaining Professional Conduct, Data recovery commands , Forensic tools: ProDiscover, SluethKit, CAINE.

[6 Hrs]

Forensic Evidence and Investigations: Functions, Categorization, Order of Volatility, Admissibility of Evidence, Acquisition and seizure of evidence, Chain of Custody, Storage formats, Multimedia Forensics: Image Capturing Process, Image Validation, Steganography, Tools for Multimedia Forensics.

[8 Hrs]

MS Windows Forensics: Windows artifacts, Program Execution artifacts, Windows Registry, Structure, Registry Analysis Tools, Taskbar Jump Lists, Automatic Destination, Custom Destination, Jump List Extract tools: Structured Storage Viewer, Windows Event Logging Service, Events Structure, Eventvwr Tool, Volume Shadow Copies, Analysis Tools, Windows Shell Bags, BagMRU keys, Prefetch Files, File Deletion, Recovery Mechanisms.

[8 hrs]

Windows File Systems: Clusters and Sectors, FAT File System, FAT Boot Sector, Interpretation using WinHex, FAT Directories, File Allocation Table, File Slack, New Technology File System (NTFS), Comparison to FAT, NTFSWalker tool, Partition Boot Sector, Boot Sector in WinHex, Master File Table (MFT), MFT File Attributes, Directory Files (Index Nodes), $INDEX_ROOT, NTFS Encrypting File System (EFS), Whole Disk Encryption, NTFS Compressed Files.

[8 Hrs]

Linux File System: Examining Linux File Structures, Ext4, Superblocks, Directory entries, Inodes, Data blocks, Acquiring file system images using dd, dcfldd, Write blocking options, Mounting images, Leveraging The Sleuth Kit (TSK) and Autopsy, fsslat, mmls, Forensic data from /etc, /usr, /var, /dev, /proc, Timeline Analysis.

[6 Hrs]

Network Forensics: Email Structure, Email Server Examination, Tracing emails, Email Forensics Tools, Cloud Forensics and Cloud Forensics Tools.

[4 Hrs]

Text Books

- “Guide to Computer Forensics and Investigations”, Bill Nelson, Amelia Phillips, Christopher Steuart, 6th Edition, Course Technology, Cengage Learning, ISBN-13: 978-1-435-49883.
- "Digital Forensic: The Fascinating World of Digital Evidences", Nilakshi Jain and Dhananjay R. Kalbande, Wiley 2016, ISBN .8126565740-978

Reference Books

- “File System Forensic Analysis”, Brian Carrier, Pearson education, 1st Edition, ISBN13:978-0321268174.
- “Handbook of Digital Forensics and Investigation”, E. Casey, Academic Press, 1st Edition, 2010, ISBN-13: 978-0123742674.
- "Cyber Forensics", Dejey and S. Murugan, Oxford University Press, ISBN 9780199489442.Dejey, Murugan, Cyber Forensics, Oxford Higher Education, 2018

(CT(DE)-22020) Digital Forensics and Data Recovery Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs/ Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Demonstrate awareness with the different evidence examination policy.
2. Perform recovery of digital evidence using a variety of software utilities.
3. Understand all aspects of evidence handling within the laboratory/industry environment.
4. Prepare for a diverse set of challenges in encountering crime scenes and identifying the optimal solution(s) to secure computer data any scenarios.

Suggested List of Assignments:

1. Study of Computer Forensics and different tools used for forensic investigation
2. Recovering Deleted Files with the help of Forensics Tools
3. Study the steps for hiding and extract any text file behind an image file/ Audio file using Command Prompt.
4. Extracting Exchangeable image file format (EXIF) Data from Image Files using Exifreader Software
5. Preparing the forensic image of the hard drive using EnCase Forensics and Restoring the Evidence Image using EnCase Forensics
6. Collecting Email Evidence in Victim PC using Forensics Tools
7. Extracting Browser Artifacts, viewing last activity of your PC, finding last connected USB on your system (USB Forensics)
8. Using Autopsy tool live forensics case investigation
   This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22012) Software Testing and Quality Assurance

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Apply fundamental concepts of software testing, process and methods.
2. Design test project, test cases, test data, test environment for conducting tests.
3. Propose software testing solutions.
4. Generate test report by making use of manual testing techniques and automation tools.
5. Devise solutions towards defects solving and software failures by making use of object-oriented and component-based software testing methods.

Course Contents

Introduction to Testing: Definition, Need of testing, Testing Principles, Basic concepts – errors, faults, defects, Failures, Verification and validation activities , Functional testing, Non functional testing, Levels of testing, Types of Testing, White box testing, Black box testing, continuous integration and continuous deployment (CICD),Regression testing, Acceptance testing, smoke and sanity testing, , Performance testing, Recovery testing, Application of statistics and probability distribution in Testing.

[7 Hrs]

Black -box testing Techniques: Need of black box testing, Black box testing Concept, Requirement Analysis, Test case design criteria, Testing methods, Requirement based testing, Positive & negative testing, Boundary value analysis, Equivalence Partitioning, Decision table testing, State transition testing, Use case testing, Cause effect graph based, Error guessing, Documentation testing & domain testing, Design of test cases.

[7 Hrs]

White-box testing Techniques: Need of white box testing, Testing types, Test adequacy criteria, Static white-box testing, Dynamic white-box testing, Structure - logic coverage criteria, Basis path testing, Graph metrics, Loop Testing, Data flow testing, Mutation Testing, Design of test cases, Testing of Object oriented systems, Challenges in White box testing.

[8 Hrs]

Test Management: Testing life cycle – Roles and activities, Test Planning –Master Test Plan-test environment, test suite, test bed, test data, test schedule, forming a test team, Develop test plan review, Test Cases design strategies, Test adequacy criteria, Coverage and control flow graphs, Paths, Loop testing, Test execution, Life cycle of defect, Defect tracking, Defect detection stages, defect types, defect severity, defect analysis and prevention.

[6Hrs]

Quality Assurance: Quality concepts – quality, quality control, quality assurance, difference between QC ad QA, Software quality assurance – SQA activities, Software reviews, Inspections, Audits, Software quality attributes - Correctness, reliability, usability, integrity, portability, maintainability, interoperability, Ishikawa's seven quality tools, six sigma.

[6 Hrs]

Testing Tools: Manual testing, manual testing Vs automated testing, automation frameworks, Automated testing tools and case studies, Study of testing tools –Test Execution tool, Selenium basics , Selenium IDE, Web-driver, Test NG, Postman, Performance, Scalability and Reliability Testing, Case studies on web based and GUI testing.

[6 Hrs]

Text Books:

- "Software Testing Techniques",Beizer B. Van Nostrand Reinhold (1990) 2nd edition, ISBN 0-442-20672-0
- "Managing the Testing Process ",Black, R. (2001) (3rd edition), John Wiley & Sons New York, ISBN: 978-0-470-40415-7.
- "Software Engineering ",Ian Sommerville , 9th Edition, Pearson, Boston, ISBN-13: 978-0-13-703515-1.

Reference Books:

- "Lessons Learned in Software Testing ",Kaner, C., Bach, J. and Petttico B. (2002), John Willey & Sons: New York, ISBN-13: 978-0471081128.
- "The Art of Software Testing Myers", Glenford J. (1979), John Wiley & Sons: New York, ISBN-13: 978-1118031964
  "The Testing Practitioner", Van Veenendaal, E. (ed.) (2004) UTN Publishers: The Netherlands, ISBN 90-72194.

Online References:

- Hetzel W. (1988) Complete Guide to Software Testing QED: Wellesley, MA.
- Lecture Notes on Basics of Statistics,Jarkko Isotalo.
  http://www.wiley.com/college/sc/reid/chap5.pdf
- Pande, Peter S., Neuman, Robert P., and Cavanagh, Roland R, "The Six Sigma Way: How GE, Motorola, and Other Top Companies are Honing Their Performance", ISBN: 0071358064.
- http://www.6sigmastudy.com/6sigmadoc/BlackBelt/1. Introduction/3. Quality Gurus 26 their contribution to Quality.pdf
- T.Pigoski,"SoftwareMaintnenance",uhcl.edu/helm/SWEBOK_IEEE/data/swebok_chapter _06.pdf
  Watts Humphrey, "Managing Software Processes" Addison-Wesley Longman Publishing Co.,Inc. Boston, MA, USA ©1989, ISBN:0-201-18095-2

(CT(DE)-22021) Software Testing and Quality Assurance Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs/ Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Devise and develop T&QA project staffing requirements.
2. Test functionality of programs using various testing techniques
3. Demonstrate use of automated testing tools.
4. Apply quality metrics in order to assure quality of software system.
5. Analyze defects in software by applying probability distribution functions.

Suggested List of Assignments

1. Develop following tests for the given program , in imperative programming language like C, C++, Java etc. :

   a. Boundary value analysis ,
   b. Equivalence Partitioning,

c. Cause effect graphing,
d. Positive & negative testing
Document Test cases and test data.

2. Develop following tests for a program by using object oriented programming language like Java :

   a. Path testing,
   b. Loop Testing,
   c. Data flow testing
   d. Mutation Testing,
   e. Coverage testing
   Document Test cases and test data.

3. Develop various test cases for website application and perform appropriate testing by applying suitable techniques on it.

4. Apply defect tracking system for a developed software.

5. Demonstrate use of automated testing tools like Selenium, TestNG etc for software developed.

6. Apply software quality assurance methods like Six Sigma for the software used in assignment 4 & 5 .
7. Perform defect analysis of a software using probability distribution for instance Rayleigh distribution.

8. Demonstrate use of Ishikawa’s tools for ensuring quality of the developed system.

9. Evaluate PSR testing depending upon the functional areas and characteristics of the problem statement / program chosen. (PSR testing is applicable for all enterprise applications).

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22013) Advanced Unix Programming

Teaching Scheme:	**Examination Scheme:**
Lectures : 3 Hrs/week	Assignment/Quizzes – 40 marks End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Correlate, describe and critically appreciate design of Unix commands, system calls and Operating System internal data structures
2. Write code snippets using system calls to emulate Unix commands, short system utilities and applications
3. Predict the output for a given code snippet consisting of various system calls

Introduction: Role of OS, CPU Protection Rings, POSIX APIs, System calls, System Call Handler, Parameter Passing, Kernel Wrapper Routines, System architecture, Files and directories, User identification, Programs and processes, Signals, POSIX, SVR4, ISOC limits, POSIX limits; sysconf, pathconf, fpathconf.

[6 Hrs]

File Management: File data structures, File Control Operations, File Status Flags, Duplicating File Descriptors, Nonblocking I/O, File Types, set-user-id, set-group-id, Sticky bit, File attributes, File Ownerships, Access Permissions,Process File Mode Creation Mask, File Size, File truncation, System data files: passwd, shadow, etc. File Systems, hard/Symbolic Links, Directory Files, File Times, Special Device Files, FIFO files, System calls: open, creat, close, lseek, read, sync, dup, fcntl, ioctl, stat, access, umask, chown, link, symlinkk, utime, getcwd.

[10 Hrs]

Process Control: Context of a process, process states and Transitions, Environment of a UNIX Process, Process Termination, Command-Line Arguments, Environment List, Memory Layout, Performing a Nonlocal Goto, Processes Identifiers, Process Creation, Termination, Monitoring Child Processes, File Sharing between Parent and Child, Zombie process, Race Conditions, Process Execution, Process Credentials, Set-User-ID and Set-Group-ID Programs, Saved Set-User-ID and Saved Set-Group-ID, Retrieving and Modifying Process Credentials, Process Time, Interpreter Files, Executing a Shell Command, system and security, Processes Accounting. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, Controlling Process,Foreground and Background Process Groups, Job Control, Shell Execution of Programs, Orphaned Process Groups, System calls: setjmp, longjmp, getrlimit, setrlimit,fork, vfork, exit, wait, exec, system, tcgetpgrp.

[12 Hrs]

Daemon Processes: Introduction, Daemon Characteristics, Coding Rules.Threads: Concepts, Identification, Creation, Termination, Synchronization, Limits, Reentrancy, Thread specific data, Thread I/O.

[4 Hrs]

Signals: Signal Types and Default Actions, Defining Signal Dispositions, Sending Signal, SIGCLD Semantics, Re-entrant functions, Displaying Signal Descriptions, alarm, Waiting for a Signal, Signal Set, Signal Mask, Pending Signals, Waiting for a Signal Using a Mask, Handling Signal with unconditional Jump, Suspending Process Execution for a Fixed Interval, Implementation of abort, system function handling signals, System calls and functions: signal, kill, raise, alarm, sigprocmask, sigpending, sigaction, sigsetjp, abort, sigsuspend, SIGCLD, sleep;

[6 Hrs]

IPC: Interprocess Communication Techniques: Pipe, popen, pclose, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores, Shared Memory, POSIX IPCs, Sockets.

[2 Hrs]

Text Books

- E W. Richard Stevens, Stephen A Rago, Advanced Programming in the UNIX Environment, Addison-Wesley, 3rd Edition, 2013, ISBN-13: 978-0-321-63773-4.
- Michael KerrisK, The Linux Programming interface, A Linux and UNIX® System Programming Handbook, 2010, ISBN-10: 1-59327-220-0.

Reference Books

- Maurice J. Bach, The Design of Unix Operating System, PHI, 2009, ISBN-13: 978-0132017992 .
- Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd Edition, O'Reilly, 2006, ISBN-10: 8184040830Terrence Chan, UNIX System Programming Using C++, Prentice Hall India, 1999, ISBN-10: 0133315622.
- Kay A Robbins and Steve Robbins, Unix Systems Programming, Pearson Education, 2004, ISBN-10: 0130424110.
- Marc J. Rochkind, Advanced UNIX Programming, 2nd Edition, Pearson Education, 2005, *ISBN*-10:0131411543.

(CT(DE)-22022) Advanced Unix Programming Laboratory

Teaching Scheme:
Laboratory : 2 Hrs/Week

Examination Scheme:
Continuous evaluation: 50 Marks
Mini Project: 20 marks
End Semester Exam: 30 Marks

Course Outcomes:

Students will be able to:

1. Develop utilities using system calls, execute and test in POSIX environments
2. Demonstrate the ability to write unix filter programs
3. Implement a shell with specified features.

List of Assignments:

1. Using the system calls for regular files
   a. Emulate the command: **cat >>file1**. Test the program by executing three times by giving different inputs to be stored in the file.
   b. Write a program to create a file with a hole: write any 10 bytes at an offset of 10 and another 10 bytes at an offset of 30. Using "system" function, invoke "od" command and view the contents. Later copy the contents of the file to another file without writing the bytes of 0. Once again verify the contents by invoking "system" with "od".
2. Using system calls on file attributes, symbolic links

   a. Create a file with permissions R, W, X for User, R, W for Group and R for Others. Display the permissions as a string. Later Remove the W from group and R from Others and set the Setuid bit on. Login as a different user and check the success of Read of access and open function.

   b. Create a list of symbolic links like e ▪ f, d ▪ e, c ▪ d, b ▪ c, a ▪ b. Input a symbolic link file name (say a). Display all the links until it is not possible to follow.

3. Using system calls on directory and file properties

   a. Does calling stat function change any of the time values? Verify with a program.

   b. Input a directory name as command line argument. Check its validity and then display all files in that directory as well as the files in all subsequent subdirectories.

4. Exercises on using system calls on exec and session

   a. This program works like a shell. Display a prompt and the user enters a command at the prompt. The command should be read from the standard input and will be executed. Do not use system API. The program will be quit when logout is entered at the prompt.

   b. Call fork. Let the child create a new session. Verify that the child becomes the process group leader and it does not have a controlling terminal

   c. Write a program to verify that when a process in an orphaned process group attempts to read() from the controlling terminal, the read() fails.

5. Exercises on using system calls on signal

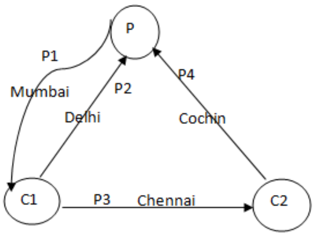
   a. Catch the SIGTERM signal, ignore SIGINT and accept the default action for SIGSEGV. Later let the program be suspended until it is interrupted by a signal.

   b. Create a child process. Let the parent sleeps of 5 seconds and exits. Can the child send SIGINT to its parent if exists and kill it? Verify with a sample program.

   c. Child inherit parent's signal mask when it is created, but pending signals for the parent process re not passed on". Write appropriate program and test with suitable inputs to verify this.

6. Exercises on using system calls on IPC – pipe

   a. A pipe setup is given below that involves three processes. *P* is the parent process, and *C*1 and C2 are child processes, spawned from *P*. The pipes are named *p*1, *p*2, *p*3, and *p*4. Write a program that establishes the necessary pipe connections, setups, and carries out the reading/writing of the text in the indicated directions.



7. Terminal Handling
    a. Write a program that calls tty_raw and terminates (without resetting the terminal mode). Then use the reset(1) command to restore the terminal mode.
    b. Write a program that calls pty_fork and have the child exec another program that you must write. The new program that the child execs must catch SIGTERM and SIGWINCH. When it catches a signal, the program should print that it did; for the latter, it should also print the terminal's window size. (This task can be extended further to have parent send signals to PTY slave).

This is an illustrative list of assignments. The instructor is expected to update the list.

(CT(DE)-22014) Embedded Systems

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Explain functioning of general purpose computing system and embedded system, architecture and features
2. Explain hardware and software design methodologies for embedded systems
3. Analyze the need for selecting embedded operating system and process of embedded system development
4. Evaluate ARM processor and system architecture along with its features and peripherals based on requirements of embedded system architecture
5. Analyze Architecture for embedded applications & recent trends of embedded systems

Course Contents

Embedded Systems Overview: Introduction, Definition, Characteristics & Salient Features, Classification, Overview of Embedded System Architecture & Recent Trends, Application Areas

[4 Hrs]

Embedded System Architecture: Hardware Architecture: Basic building blocks, Devices and Communication Interfaces, Software Architecture: Embedded Operating System, Applications Software & Communication Software (Firmware)

[6 Hrs]

Embedded Operating System: General Purpose Operating Systems vs Embedded Operating Systems, Overview of Embedded Operating Systems, detailed study of Real Time Operating

System(RTOS) such as TI RTOS, Task Scheduling, Task Synchronization, Inter- task communication concepts on Embedded TI RTOS platform

[8 Hrs]

Embedded System Development: Development Process, Requirement Engineering, Design, State Transition Diagram, Implementation, Integration and Testing, Cross Platform Development, Process of generating Single Executable Image of Firmware, Boot Sequence, Hardware and Software Development Tools. Case Studies of Typical Embedded Systems : Digital Camera and Washing Machine.

[7 Hrs]

ARM processor: Introduction ARM architecture and Cortex : M series, Introduction to the Tiva family viz. TM4C123x and its targeted applications, Tiva block diagram, address space, on-chip peripherals, Simplified view: block diagram, programming model: Registers, Operation modes, Exceptions and Interrupts, Reset Sequence

[8 Hrs]

ARM Cortex M4 Peripherals: I/O pin multiplexing, GPIO control, Watchdog Timer, System Clocks and control, Hibernation Module on TM4C, Introduction to Interrupts, Interrupt vector table, interrupt programming, Timers, ADCs, UART and other serial interfaces: USB, PWM, RTC, DMA, QEI on TM4C microcontroller

[7 Hrs]

Text Books

- K.V.K. Prasad, Embedded / Real Time Systems: Concepts, Design and Programming Black Book, Dreamtech Press, 2005 publications ISBN 10: 8177224611 / 13: 9788177224610
- Joseph Yiu's The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors – Third Edition, published by Elsevier ISBN: 9789351071754, 9351071758
- Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers, 2014, Jonathan W Valvano Createspace publications ISBN: 978-1463590154.
- Embedded Systems: Introduction to ARM Cortex - M Microcontrollers, 5th edition Jonathan W Valvano, Createspace publications ISBN-13: 978-1477508992
- Vahid F. and Givargies T., Embedded Systems Design, John Wiley X. Sons, 2002, Third edition, ISBN:978-81-265-0837-2

Reference Books

- John B Peatman, Design with PIC Microcontrollers, Pearson Education, 1998, ISBN:978-81-7758-551-3
- Liu, Real-Time Systems, Pearson Education, 2000. 5. Technical Manuals of ARM Processor Family available at ARM Website on Net
- Doug Abbott, Linux for Embedded and Real-time Applications, Newnes, Fourth Edition, ISBN: 978-0128112779

- Raj Kamal: Embedded Systems – Architecture: Programming and Design: McGraw Hill Education, Second Edition, ISBN:978-0070667648

(CT(DE)-22024) Embedded Systems Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Write and demonstrate a working solution for a given problem
2. Think of all possible inputs to an application and handle all possible errors properly
3. Develop a system using the concepts of RISC architecture and ARM processors
4. Write software programs for ARM processor to develop a suitable application

Suggested List of Assignments

List of Experiments based on Cortex M4 Tiva Microcontroller:

A. Basic Interfacing experiments using Tiva
   1. Blink an LED.
   2. Blink an LED using delay generated by Timer.
   3. Generating PWM using the in-built PWM module.
   4. Configuring PWM and ADC to control speed of DC motor
   5. Echo back the data using the UART module

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22015) Natural Language Processing

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs / Week	T1 and T2 : 20 marks each
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Demonstrate the understanding of basic text processing techniques in NLP.
2. Design, implement and evaluate part-of-speech taggers and parsers for a language.
3. Build language models for different languages.
4. Apply knowledge base for Word Sense Disambiguation.
5. Analyze and build word embeddings for different languages.

Course Contents

Introduction: What is NLP, Fundamental and Scientific goals, Engineering goals, stages of NLP, problems in NLP, Applications of NLP, Empirical Laws of language, zipf's law, Heap's law.

[4 Hrs]

Basic Text Processing and Morphology: Tokenization, word token, word type, sentence segmentation, feature extraction, issues in tokenization for different languages, word segmentation, text segmentation, normalization, case folding, Spelling Correction, Morphology-morphemes, types, Inflectional and Derivational morphology, , Stemming, lemmatization, Porters Algorithm, , Spelling correction -minimum edit distance, N-gram Language Modeling- probabilistic language model, auto completion prediction, Evaluation and perplexity, Smoothing techniques.

[8 Hrs]

POS Tagging : Sequence labeling tasks of NLP, POS tagging, POS tag sets, Hidden Markov Model-Introduction, Markov Processes, HMM characterization -Likelihood of a sequence (Forward Procedure, Backward Procedure), Best state sequence (Viterbi Algorithm), Re-estimation(Baum-Welch - Forward-Backward Algorithm) , Models for Sequential tagging – Maximum Entropy, Conditional Random Field, Long Short Term Memory (LSTM).

[8 Hrs]

Syntactic Parsing : Constituency and dependency parsing, Constituency parser -Syntactic structure, Parsing methodology, Different parsing algorithms, Parsing in case of ambiguity, CKY algorithm, Issues in parsing, Probabilistic parsing –PCFG, inside-outside probabilities, Dependency parsing- Syntactic structure, Parsing methodology, Transition-Based Dependency Parsing, Graph-Based dependency parsing, Evaluation, Co-reference resolution, Named-entity recognition.

[10 Hrs]

Lexical Semantics : Introduction, WordNet: Word Senses, Word relations, Word similarity and thesaurus methods, Word Sense Disambiguation, Novel Word Sense Detection, Applications.

[6 Hrs]

Distributional Semantics : Distributional Hypothesis, Vector Space Model for Semantics, One hot encoding, Distributional Representation, tf-idf, Point wise Mutual Information, Word Vectors, Vector offsets for Analogy Reasoning, Word embeddings, Methods of generating word embeddings: Skip-gram, CBOW, Glove model, Soft-max, evaluation measures-rough scores, Applications.

[6 Hrs]

Text Books

- Daniel Jurafsky and James H. Martin, "Speech and Language Processing", Second Edition, Prentice Hall, 2008, ISBN: 978-0131873216.
- Allen James, "Natural Language Understanding", Second Edition, Benjamin/Cumming, 1994, ISBN: 978-0805303346.

- Chris Manning and Hinrich Schuetze, "Foundations of Statistical Natural Language Processing", MIT Press, ISBN: 978-0262133609.

Reference Books

- Journals: Computational Linguistics, Natural Language Engineering, Machine Learning, Machine Translation, Artificial Intelligence.
- Conferences: Annual Meeting of the Association of Computational Linguistics (ACL), Computational Linguistics (COLING), European ACL (EACL), Empirical Methods in NLP (EMNLP), Annual Meeting of the Special Interest Group in Information Retrieval (SIGIR), Human Language Technology (HLT)

(CT(DE)-22023) Natural Language Processing Laboratory

Teaching Scheme:	**Examination Scheme:**
Laboratory :2 Hrs/Week	Continuous evaluation: 50 Marks End Semester Oral Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Demonstrate the understanding of basic text processing techniques in NLP.
2. Build language models.
3. Build parts of speech tagger and dependency parsers for English language.
4. Apply WSD technique using knowledge base WordNet for English language.
5. Study and apply word embeddings to real-life problems.

List of Assignments:

Please describe as tasks to be performed. More details required.

1. Social Media Data Tokenization & Normalization.
2. Designing spell checking program.
3. Language Modeling.
4. Viterbi Decoding Algorithm for Parts of Speech tagging.
5. LSTM for Parts of Speech tagging.
6. Design and implement a co-reference resolution model.
7. Building a dependency parser for given input sentence.
8. Word sense disambiguation using WordNet.
9. Studying word embeddings for English/Hindi/Marathi language.
10. Build an application using word embedding to solve real-time problem.

(CT(DE)-22016) Parallel Computer Architecture and Programming

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Justify the need of high performance computing.
2. Demonstrate quantitative design principles of parallel computer architectures.
3. Measure and analyze performance through different benchmarks and utilities.
4. Differentiate various parallel computer architectures and programming models.
5. Demonstrate and build a basic cluster setup.

Course Contents

Fundamentals of Performance oriented Architecture Design: Defining Computer Architecture, Trends in Technology, Trends in Cost, Dependability, Measuring, Reporting and Summarizing Performance, Quantitative Principles of Computer Design, Moore’s Law, Amdahl’s Law, Gustafson’s Law, Flynn’s Classification of Computer Architectures, Recent Computing Trends, Top 500 Ratings, Fundamentals of Computer Design, Basic and Intermediate concepts of pipelining , Pipeline Hazards, Pipelining Implementation issues.

` **[5 Hrs]**

Instruction-Level Parallelism and its Exploitation: Concepts and Challenges, Basic Compiler Techniques for Exposing ILP, Reducing Branch Costs with Prediction, Scheduling, Overcoming Data Hazards with Dynamic Scheduling, Dynamic Scheduling: Algorithm and Examples, Hardware-Based Speculation, Studies of the Limitations of ILP, Limitations on ILP for Realizable Processors, Hardware versus Software Speculation, ILP Support to Exploit Thread-Level Parallelism.

[8 Hrs]

Data-Level Parallelism in Vector, SIMD: Vector Architecture, Vector Instructions, AVX support, Vectorizing code, SIMD Instruction Set Extensions for Multimedia, Detecting and Enhancing Loop-Level Parallelism, Case study of vector architecture

[7 Hrs]

Memory Hierarchy Design: Basics of Memory Hierarchy, Cache Performance, Basic Cache Optimizations and numericals, Shared and Private Cache, Virtual Memory, Protection and Examples of Virtual Memory, Memory Technology and Optimizations, The Design of Memory Hierarchies, Study of Memory Hierarchies in different Architectures, Case studies.

[7 Hrs]

Thread-Level Parallelism: Introduction to Shared Memory Architectures, Loosely and Tightly coupled multiprocessors, Centralized Shared-Memory Architectures, Snoopy Bus Cache Coherence, Performance of Shared-Memory Multiprocessors, Distributed Shared Memory and Directory Cache

Coherence, Basics of Synchronization, Models of Memory Consistency, Examples of Cache Coherence and Consistency.

[7 Hrs]

Parallel Programming Paradigms: Cluster and Network of Workstations (COW and NOW), Different ways of building a cluster, Parallel Programming Models: Shared Memory, Message Passing, Data Parallel, MPI/PVM, Parallel Algorithm examples: Matrix Multiplication, Sorting, Introduction to Parallel Programming Languages. Case studies of different cluster/ server architectures.

Warehouse-Scale Computers: Architecture, Programming Model and Workloads

[6 Hrs]

Text Books

- John L Hennessy, David A Patterson, "Computer Architecture: A Quantitative Approach", Fifth Edition, Morgan Kaufmann, 2011, ISBN-13: 978-8178672663
- Kai Hwang, Naresh Jotwani, "Advanced Computer Architecture", Third Edition, Tata McGraw-Hill Edition, 2016, ISBN: 978-9339220921.

Reference Books

- D. E. Culler, J. P. Singh, and A. Gupta, "Parallel Computer Architecture", Second Edition, Morgan Kaufmann, 2017, ISBN: 978-1-4987-7271-6.
- Hesham El-Rewini, Mostafa Abd-El-Barr, "Advanced Computer Architecture and Parallel Processing", Wiley, 2005, ISBN: 9780471467403

(CT(DE)-22025) Parallel Computer Architecture and Programming Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks
	End Semester Oral Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Learn basics of quantitaive approaches to analyze performance of computing systems.
2. Demonstrate performance statistics using perf utility.
3. Implement programs using different parallel programming paradigms.
4. Implement message passing programs in distributed environment.
5. Demonstrate the different steps involved in building of a simple cluster.

Suggested List of Assignments

1. Study of different benchmark suits used to evaluate performance of different systems.
2. Performance statistics observation using perf utility.

3. Program to execute matrix multiplication using pthreads.
4. Program to execute matrix multiplication using OpenMP and comparison with pthread program.
5. Program to execute Pi computation and prefix sum using OpenMP.
6. Program to execute section, task and synchronization constructs of OpenMP.
7. Case Study of Cluster building steps - MPI Cluster setup and overview of different routines.
8. Program to implement point to point communication using MPI routines.
9. Program to implement collective communication using MPI routines.
10. Program to implement Map-Reduce parallelism for Warehouse Scale Computer.

Reference Books

- Peter S. Pacheco, "An Introduction to Parallel Programming", Morgan Kaufmann, Morgan Kaufmann, 2011, ISBN: 978-0-12-374260-5.
- Michael Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill Edition, 2003, ISBN13: 978-0072822564.

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22017) Cyber Security

Teaching Scheme:	**Examination Scheme:**
Lectures : 3 Hrs/week	Assignment/Quizzes – 40 marks End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Define the need of Cyber Security.
2. Explain the IT act, Application Security vulnerabilities and its mitigation techniques.
3. Demonstrate the knowledge of penetration testing, and social networking security.
4. Analyse the malwares, social networking websites and impact of cyber-crime on e-commerce.

Unit I:Introduction: Nature and scope of computer crime, Understanding how cyber criminals and hackers work, Different types of cyber-crimes, Introduction to digital signatures, Cryptography, Digital certificate and public key infrastructure, IT Act., Impact of cyber-crime on e-governance and e-commerce.

[6 Hrs]

Unit II: Malware reverse engineering: Overview of malware reverse engineering, Types of malware, Malicious code families, Latest trends in malware analysis, Basic static and dynamic analysis, Malware analysis techniques, Case study.

[6 Hrs]

Unit III:Web application security: Introduction to web application security: Attacks, vulnerabilities and mitigation, Client-side security, Server-side security, Application security: HTTPS, HSTS etc., Security engineering: Passwords and their limitations, Attacks on passwords: CAPTCHA, OTP.

[8 Hrs]

Unit IV:Advanced security topics: Secure email systems: PGP, SMIME, DKIM, DMARC, DNSSec, SMTP STS etc., Privacy and security for online social networks, Database security, Browser security, Mobile device security.

[8 Hrs]

Unit V: Ethical hacking and penetration testing: Security Technologies: IDS, IPS, Ethical hacking, Penetration testing fundamentals: Reconnaissance, scanning, gaining access, maintaining access, Covering tracks.

[6 Hrs]

Unit VI:Case studies: Cloud security, Operating system security, Security of social networking websites, IoT devices security, E-commerce websites security.

[6 Hrs]

Text Books

- Hossein, "Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management", Wiley, Volume 3 edition, ISBN-13: 978-0470323069.
- Georgia Weidman, "Penetration testing: A Hands-On Introduction to Hacking", No Starch Press, 2014, ISBN-13: 978-1593275648.
- Michael Sikorski and Andrew Honig, " Practical Malware Analysis", No Starch Press, 1st Edition, 2012, ISBN-13: 978-1593272906

Reference Books

- "Practical Internet of Things Security" by Brian Russell, Drew Van Duren, Packt publishing, 2016, ISBN: 9781785889639
- T. Mather, S. Kumaraswamy, S. Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly Series, 2009, ISBN-13: 978-0596802769.
- "Cyberlaw: the Indian perspective"; Pavan Duggal; Saakshar Law Publications, 1st edition, 2002, ISBN: 8189121022, 9788189121020.

(CT(DE)-22026) Cyber Security Laboratory

Teaching Scheme:	**Examination Scheme:**
Laboratory : 2 Hrs/Week	Continuous evaluation: 50 Marks
	Assignments/Mini Project: 20 marks
	End Semester Exam: 30 Marks

List of Assignments:

1. Perform literature survey of recent research papers on cyber security.
2. Perform case study of any two cyber-crime cases in India and write a report illustrating Indian cyber laws (IT Act, IT Act 2008. IPC) relevant to these cases.
3. Perform malware reverse engineering in an isolated environment using sandbox and document the findings.
   a. Use any sample malware file.
   b. Study malware behaviour: its working, how it spreads. Find its features/characteristics.
   c. Identify the changes made by malware on the system. (for e.g., changes in event log, registry etc.). Difference between infected and normal system.
   d. Describe detection method or alert system for malware
   e. Write down steps to remove the malware and make system safe again.
4. Perform penetration testing using Kali Linux on virtual machine and write a report.Follow each stage of penetration testing: Planning and reconnaissance, scanning, gaining access, maintaining access and analysis.Once the Kali is installed, use following tools. Submit a report answering following questions:
   a. Maltego: How did you perform the following things?
   - Associate an e-mail address to a person
   - Associate websites to a person
   - Verify an e-mail address
   - Gather details from Twitter, including geolocation of pictures

   b. Vega: Provide a target website and scan that website for vulnerabilities.
   c. NMAP: Scan your local network and provide screenshot of the report
   d. Tamper Data plugin in FireFox: Gather information of GET and POST request for gmail.com
   e. Metasploit framework: List out all the exploits provided by metasploit framework. Explain preconditions and expected end results for each one.
5. Provide screenshots also. Do not use any exploit without proper permission.
6. Implement any IDS/IPS system.

This is an illustrative list of assignments. The instructor is expected to update the list.

Departmental Elective – IV

(CT(DE)-22027) System Administration

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs./week	Assignment/Quizzes: 40 marks End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Carry out the following tasks, with special emphasis on GNU/Linux based systems:

2. Install various Linux distributions and Windows-based servers and desktop systems on commodity hardware, and carry out basic administration tasks of the user, network, process, storage management.
3. Setup a LAN-based laboratory using DHCP.
4. Install and configure a web server, database server, DNS, NFS, NIS, LDAP based system, secure a Desktop and a server system completely using existing tools.
5. Learn how to manage the Linux file system and the storage of data
6. Provide security administration for Linux

Course Contents

Basic System Administration: Partitioning, Installation of multiple operating systems on desktops, Various operating system services: Cron, CPU utilization, User management, Backup, Log management, Boot loader, Process management, File system namespace, Kernel upgrades.

[8Hrs]

Network Administration: Configuration of network hardware, Linux router, Managing routes, DHCP - Dynamic Host Configuration Protocol, DNS-Domain Name Server, NFS - Network File System, NIS - Network Information Service, Email Setup-SendMail, Issues with mail services, POP and IMAP Server.

[8Hrs]

Filesystem Administration: Formatting, Partitioning, Managing file systems, Defragmentation, Quotas, Journaling file system, Logical volume management, Disk layouts, File system check, Archiving and compressing files, SAN, NAS, Case Studies: ext2, ext4, ntfs, samba, cifs, lvm, fat32.

[8Hrs]

Security Administration: Methods of attack, Network security tools- Nmap, Snort, Nessus, Wireshark/tcpdump, Firewall, IPtables, NAT, IP filtering, Setting up linux firewall, Mandatory access control (MAC), SELinux, Cryptographic security tools Authentication mechanisms, LDAP, Proxy servers.

[8Hrs]

Server Administration: Apache webserver configuration, Database servers: MySQL and PostgreSQL

[4Hrs]

Devices Administration: Device resources, udev: Device files, Installing and configuring printers, Managing printers via the web interface, Scanners, PCI devices, LAN cards, Device troubleshooting, Plug and Play devices.

[4Hrs]

TextBooks

- Evi Nemeth, Garth Snyder, Ben Whaley, Trent R. Hein, "UNIX and Linux System Administration Handbook", Pearson Education, Fourth edition, ISBN-13: 978- 8131761779
- Richard Petersen, "Linux: The Complete Reference, Sixth Edition", Mcgraw-Hill Education Sixth edition,ISBN-13: 978-0070222946

- Arnold Robbins, Nelson H. F. Beebe, "Classic Shell Scripting", Shroff/O'ReillyFirst edition, ISBN-13: 978-8173668463
- Wale Soyinka, "Linux Administration: A Beginner's Guide", McGraw-Hill Osborne Media Publication Sixth Edition, ISBN-13: 978-0071767583
- Olaf Kirch & Terry Dawson, "Linux Network Administrator's Guide", O' Reilly 2nd Edition June 2000,ISBN-10: 1565924002,ISBN-13: 978-1565924000
- W.Preston, "Using SANs and NAS",O' Reilly; First Edition, February 2002,ISBN-10: 0596001533,ISBN-13: 978-0596001537

Reference Books

- Richard Blum, Christine Bresnahan, "Linux Command Line and Shell Scripting Bible", Wiley India Pvt. Ltd., Second edition, ISBN-13: 978-8126533831;
- Jean-Paul Tremblay, Paul. G. Soresan, "An introduction to data structures with Applications", Tata Mc-Graw Hill International Editions, 2nd edition 1984, ISBN-0-07- 462471-7

(CT(DE)-22033) System Administration - Laboratory

Teaching Scheme	Examination Scheme
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Learn how to install the Linux operating system and software.
2. Configure and setup the LAN
3. Do Configuration and installation of linux based ssh server, telnet server, web server, database server
4. Manage users and groups by adding/deleting/modifying, configuring LDAP
5. Manage system storage by using partitions, logical volumes and phsyical volumes
6. Understand and configure SELinux in Linux

Suggested List of Assignments

1. Set a desktop with following software configuration options: triple boot with Windows, Ubuntu and Fedora operating systems; Grub timeout set to 5 seconds with default Fedora Linux; Each step of boot process secured with passwords; Following software installed in each OS: office, internet browser, c compiler, terminal, ssh server, telnet server, web server, database server.
2. Set up a LAN based computer laboratory with 5 computers - using DHCP first and then using static IPs; one of the machines should work as DHCP server; set up LDAP+NFS based authentication for managing single identity on all computers; setup quotas on NFS systems.

3. Demonstrate use of three-tier architecture using LAMP and WAMP suite using same code base.
4. Setup a disk management system using LVM such that all options of LVM are demonstrated.
5. Write a program to browse an ext4 file system and locate data of a deleted file.
6. Setup a proxy server and firewall with set of policies to block video content in the network.
7. Setup apache web server to serve 3 websites, with 3 domain names at a time from a single machine; demonstrate the use of at least 5 configuration options of apache.
8. Setup an email server and demonstrate use of at least 3 email clients to send email using it.
9. Setup SE Linux.

The instructor is expected to update the list

(CT(DE)-22028) Storage & Virtualization

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs./week	Assignment/Quizzes: 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Apply, compare and contrast basic concepts of hardware storage, NAS-SAN, virtualization concepts.
2. Analyze cloud computing setup with its vulnerabilities and applications using different architectures.
3. Assess cloud Storage systems and cloud security, the risks involved, its impact and develop suitable cloud application.
4. Design and implement Virtualization environment and apply cloud resource management.
5. Manage different storage technologies

Basics of Systems and Storage: Storage Challenges, Issues- Data sources, challenges of data growth and data availability, Performance and manageability requirements, Data virtualization, OS and device drivers - Kernel, firmware, RDMA, boot sector, device partitioning, UNIX file system, virtual memory, namespace, metadata, buffer cache, defragmentation.

[6 Hrs]

Storage Hardware and SCSI Protocol: Storage hardware and SCSI Protocol, SAN, NAS, Fibre Channel and iSCSI, Storage Hardware Building Blocks - Device types, HBA, switches, hubs, routers, GBIC, Introduction to various Storage protocols, Overview of IDE, SAS, SATA, SCSI, FC, FCoE, iSCSI, Infiniband, FCP, FC-IP, iFCP, Fibre Channel Protocol Stack & Concepts- FC ,Mapping Protocols - iSCSI, FCP- SCSI mapping to underlying transport, Connection Management, PDU,TOE.

[6 Hrs]

SAN & NAS: SAN concepts, DAS, SAN architecture, concepts of zoning, name server, SCN, WWN, routing, FC-SAN, IP-SAN and Applications, NAS concepts, NAS architecture, Protocols (CIFS, NFS), Performance, Scalability and Usability, Appliances.

[6 Hrs]

Storage Virtualization Concepts: Data Centre end-to-end view, Overview of stack, clustering, virtual Machines, cloud storage and virtualization, RAID levels, I/O stack, OS abstraction, Storage Pooling, Storage Provisioning, Online Grow/Shrink Storage Virtualization, Metadata management, Transaction consistency, I/O maps, I/O path considerations, Data consistency, Crash recovery, Application interfaces, Linux container.

[7 Hrs]

Applications and Use Cases for Storage Virtualization: Storage Virtualization use cases, applications, data replication, RPO/RTO, Snapshots, Data protection, Restore, Archival, Compliance considerations. Capacity Management, Storage provisioning, De-duplication, thin provisioning, Storage Tier, ILM, Data classification, Storage grid.

[7 Hrs]

Cloud Computing: Virtualized environments features, taxonomy of virtualization techniques, virtualization, Cloud Computing basics, Cloud architectures, Services and applications, Cloud Computing mechanisms, MapReduce programming model, Benefits and challenges, Cloud resource management and scheduling, Cloud migration, Service management and SLAs, Security, and monitoring, Cloud Platforms in industry, Case study of AWS, Google Cloud, Microsoft Azure and IBM Softlayer.

[8 Hrs]

Text Books

- Storage Networks: The complete Reference. Robert Spalding TMH.
- Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs, Second Edition Publisher: Addison-Wesley Author: Tom Clark.

Reference Books

- Cloud Computing: Concepts, Technology & Architecture, Thomas Erl, Prentice Hall, 2013.
- Cloud Computing: Principles and Paradigms, Wiley Series on Parallel and Distributed Computing, Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, John Wiley and Sons, 2011,ISBN 978-0-470-88799-8.
- Cloud Computing: Theory and Practice, Dan C Marinescu, Elsevier, 2013.
- Cloud Computing Bible, Barrie Sosinsky, Wiley Publishing, 2011. · Virtualization Essentials , Matthew Portnoy, John Wiley and Sons, 2012.
- Computer Systems - A Programmer"s Perspective, Randall Bryant and David O"Hallaron Pearson Education, 2003.
- The Design and Implementation of the 4.4 BSD Operating System, McKusick, Bostic, Karels, Quaterman, 1996.

(CT(DE)-22034) Storage & Virtualization - Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Configure remote data replication functionality.
2. Implement a simple file system.
3. Demonstrate the concept of creating a snapshot of a hard drive.

Suggested List of Assignments

1. Create synchronous remote replication functionality from host1 to host2. Implement the functionalities :
   a) Data written on local disk should be synchronously copied to remote disk
   b) A write from an application should be completed only when data has been written to both source and replicated node.
2. Writing a simple File System which provide functionality of "ls, mkdir, pwd" posix commands. Apart from that it should support print operation. Write an application programme to test these functionality.
3. Using the hard disk on one host create a snapshot for the same and store in a flat file on other m/c. Provide a mechanism to restore the data from flat file.
4. Write a "Examination data server" using MySQL open source database for storing students examination report, which is a snapshot of all present student's report of college. The report consist of six field: Name, Id, Branch, Grade {in each semester}, Final Grade, Backlog Courses. Write a client program to query on various parameters. Server should generate a report for same {e.g. The no of students having CGPA (final grade) 7.0 and above in all branches/particular branches. The No. of students not cleared in Mathematics yet, Name of topper in every branch etc..}. Use pthreads to support multiple client queries. Also implement write functionality {update in case of old students / addition (new students) / deletion (final semester student who has passed out} and use reader/writer lock.

   This is an illustrative list of assignments.

   The instructor is expected to update the list.

(CT(DE)-22029) Mobile and Ad-Hoc Networks

Teaching Scheme:	**Examination Scheme:**
Lectures : 3 Hrs/week	Assignment/Quizzes – 40 marks End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the current topics in MANETs and WSNs, both from an industry and research point of views.
2. Demonstrate the principles of mobile adhoc networks (MANETs) and what distinguishes them from infrastructure-based networks.
3. Analyze how proactive routing protocols function and their implications on data transmission delay and bandwidth consumption.

Course Contents

Introduction:
Fundamentals of wireless communication technology, Characteristics of the wireless channel, multiple access techniques, IEEE 802 networking standard, Wireless LANs and PANs, IEEE 802.11 standard, IEEE 802.16 Standard, Bluetooth, HomeRF.

[6 Hrs]

Cellular Wireless Networks And Wireless Internet:
The cellular concept, Cellular architecture, Generations of cellular systems, Wireless in local loop, Wireless ATM, Wireless internet, What is wireless internet, Mobile IP, TCP in wireless domain, WAP, Optimizing web over wireless.

[8 Hrs]

Introduction To Ad-Hoc Networks:
Characteristics, applications, Issues in Ad Hoc wireless networks medium access protocols: design issues, Goals and classification, Contention based protocols- with reservation, Scheduling algorithms, Protocols using directional antennas, IEEE Standards: 802.11a, 802.11b, 802.11g, 802.15, HIPERLAN

[6 Hrs]

Routing Protocols:
Design issues, Goals and classification. Proactive Vs reactive routing, Unicast routing algorithms, Multicast routing algorithms, hybrid routing algorithm, Energy aware routing algorithm, Hierarchical routing, QoS aware routing.

[8 Hrs]

Other Features:
Issues in designing transport layer Ad-Hoc networks, TCP over wireless Ad-Hoc networks, Security issues in Ad-hoc networks, Secure routing protocols, Energy management in Ad-Hoc networks, Battery management, Transmission power management and system power management.

[6 Hrs]

Wireless Sensor Networks:
Architecture, Data dissemination, Data gathering, MAC protocols for sensor networks, Location discovery, Quality of a sensor network, Physical design of IoT, Logical design of IoT, IoT enabling technology

[6 Hrs]

Text Books

- C. Siva Ram Murthy and B. S. Manoj, Ad hoc Wireless Networks Architectures and protocols, 2nd edition, Pearson Education. 2007

Reference Book

- F.Zhao, L. Guibas, Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, 2004
- Stefano Basagni, Marco Conti, Silvia Giordano and Ivan sSojmenovic, Mobile Ad-hoc Networking, Wiley-IEEE Press, 2004.
- Mohammad Ilyas, The Handbook of Ad-hoc Wireless Networks, CRC press, 2002.

(CT(DE)-22035) Mobile and Ad-Hoc Networks Laboratory

Teaching Scheme:	**Examination Scheme:**
Laboratory : 2 Hrs/Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Apply knowledge of wireless Ad-hoc networks to various application areas.
2. Design, implement and maintain wireless Ad-hoc networks.
3. Formulate and solve problems creatively.
4. Practical knowledge acquired by hands-on session.

List of Assignments:

1. Analyze how the Data Rate of a Wireless LAN (IEEE 802.11b) network varies as the distance between the Access Point and the wireless nodes is varied
2. Analyze the working and routing table formation of Interior routing protocols, i.e. Routing Information Protocol (RIP) and Open Shortest Path First (OSPF)
3. Analyze the performance of a MANET, (running CSMA/CA (802.11b) in MAC) with increasing node mobility
4. Develop and Analyze the performance of TCP connection when it is used for wireless networks. You will find performance of TCP decreases dramatically when a TCP connection traverses a wireless link on which packets may be lost due to wireless transmission errors. Make use of Active Queue Management Technique to control congestion on Wireless Networks. Evaluate the performance of FIFO, RED and WFQ over wireless networks using suitable Network Simulator.
5. Develop unicast routing protocols using any suitable Network Simulator for (Mobile Ad hoc Networks) MANET to find the best route using the any one of routing protocols from each category from table-driven (e.g., link state or DSDV) on demand (e.g., DSR, AODV, TORA), hybrid (e.g., ZRP, contact-based architectures) and hierarchical (e.g., cluster based.) The efficient path/route should be established for source and destination data transmission using

routing protocols. Understand the advantages and disadvantages of each routing protocol types by observing the performance metrics of the routing protocol. In that way the best application/environment suitable routing protocol can be identified in each category.

6. Develop multicast routing protocols using any suitable Network Simulator for MANET in which session nodes are connecting through either tree(MAODV, MCEDAR) or mesh (ODMRP, CAMP, FGMP) structure. Analyze the performance metrics of multicast routing protocols with unicast routing protocols.

This is an illustrative list of assignments. The instructor is expected to update the list.

(CT(DE)-22030) Computational Biology

Teaching Scheme:	**Examination Scheme:**
Lectures : 2 Hrs/week	Assignment/Quizzes – 40 marks
Tutorial : 1 Hrs/ week	End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Demonstrate the concept of information generation from protein sequences, DNA sequences, whole genome.
2. Extract information from large databases and to use this information in computer modeling.
3. Evaluate various elements of computational biology such as genomic networks, algorithms, and models
4. Design and develop current applications of computational biology.

Introduction: Molecular Biology Introduction, Cell, Nucleus, Genes, DNA, RNA, Proteins, And Chemical structure of DNA, RNA, Transcription and Translation Process. Protein Structure and Functions, Nature of Chemical Bonds Molecular Biology tools, Polymerase chain reaction

[8 Hrs]

Sequence Alignment: Simple alignments, Gaps, Scoring Matrices, Global and Local Alignments, Smith-Waterman Algorithm, Multiple sequence Alignments, Gene Prediction, Statistical Approaches to Gene Prediction

[5 Hrs]

Genome Algorithms: Genome Rearrangements, Sorting by Reversals, Block Alignment and the Four-Russians Speedup, Constructing Alignments in Sub-quadratic Time, Protein Sequencing and Identification, the Peptide Sequencing Problem

[8 Hrs]

Microarray Data Analysis: Microarray technology for genome expression study, Image analysis for data extraction, Data analysis for pattern discovery, gene regulatory network analysis

[5 Hrs]

Phylogenetic: Neighbor's relation method, Neighbor-joining method, Maximum likelihood Approaches, Multiple Sequence Methods Structural Biology, Sequence, organisms, 3D structures, complexes, Assemblies, Case Studies, examples

[8 Hrs]

Drug Discovery & Next Gen Sequencing: Similarities/differences between drugs and receptors, protein-ligand docking, Massively Parallel Signature Sequencing (MPSS), SOLiD sequencing, Single molecule real time (SMRT) sequencing

[6 Hrs]

Text Books

- Dan E. Krane, Michael L. Raymer, "Fundamental Concepts of Bioinformatics,", Pearson Education, Inc. Fourth Edition, 9780805346336.
- Harshawardhan P. Bal, "Bioinformatics Principles and Applications", Tata McGraw-Hill, seventh reprint, 9780195692303.

Reference Books

- Teresa Attwood, David Parry-Smith, "Introduction to Bioinformatics", Pearson Education Series, 9788180301971
- R. Durbin, S. Eddy, A. Krogh, G. Mitchison., "Biological Sequence Analysis: Probabilistic Models of proteins and nucleic acids", Cambridge University Press, 9780521629713.

(CT(DE)-22036) Computational Biology Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Implement string matching algorithms
2. Implement solutions for sequential alignment problems
3. Solve phylogeny, topology, related problems
4. Use the tools for computational biology

Suggested List of Assignments

Study any biological database freely available on Internet; prepare a detailed report of it.

1. Implement string matching algorithm from available ones. Find time & space complexity.
2. Design and implement Smith-Waterman pairwise sequence alignment algorithm using any language (C, C++, Java, python, etc.) and test it for some arbitrary sequences of Protein.

3. Design and implement Needleman-Wunsch pairwise sequence alignment algorithm using any language (C, C++, Java, python, etc.) and test it for some arbitrary sequences of Protein.
4. Prepare detail report on Bioinformatics tools other than BLAST and FASTA Assignment
5. Solve computational problems for reconstructing the phylogeny: Character based and distance based.
6. Write Program to find the parsimony length of a given tree topology.
7. Find multiple sequence alignment using ClustalW2.
8. Perform data mining using Clementine on any arbitrary sequences.
9. Prepare detail report on Case study: A model consists of a rooted tree which models the evolution relationship (Cavender-Felsenstein model and Jukes-Cantor model).

This is an illustrative list of assignments. The instructor is expected to update the list.

(CT(DE)-22031) Multicore Technology

Teaching Scheme:	**Examination Scheme:**
Lectures : 3 Hrs/week	Assignment/Quizzes – 40 marks
	End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Learn & analyze the working principles of multicore architectures.
2. Optimize performance of multicore systems.
3. Specify the necessity of SMP & AMP.
4. Comprehend and differentiate between SMP & AMP.
5. Identify and demonstrate the need of domain specific architectures.

Course Contents

Introduction to Multicore Systems: Fundamentals, The Era of Multicore Machines, Unicore vs Multicore - Understanding Performance, Shared memory Multicore Systems, Distributed Memory Multicore Systems, Hybrid Systems Symmetric and Asymmetric Multicore Systems, Overview of Multithreading, Multithreading in different forms, Homogeneous and Heterogeneous Multicore systems, Examples of different Multicore Systems.

[6 Hrs]

Cache Memory: Large Cache Design: Shared vs. Private Caches, Centralized vs. Distributed, Shared Caches, Coherence: Snooping-based cache coherence protocol, directory-based cache coherence protocol, Uniform Cache Access, Non-Uniform Cache Access, S-NUCA, D-NUCA, Inclusion, Exclusion, Examples of different Cache Organization, Consistency Models, Case Study.

[7 Hrs]

Performance and Optimizations for Multicore Systems: Select the right "core" - Improve serial performance - Achieve proper load balancing - Improve data locality - Reduce or eliminate false sharing - Use of affinity scheduling - Lock granularity and frequency - Remove synchronization barriers - Minimize communication latencies - Use of thread pools - Managing thread count - Use of parallel libraries.

[6 Hrs]

Programming Multicore Systems: Programming models for Multicore Systems – Shared Memory Programming using pthreads - Shared Memory Programming using OpenMP – Use of OpenMP compiler directives – #pragma with different clauses – Understanding parallelized loops – Synchronization Constructs towards dependencies – Function parallel program - OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, Performance.

[8 Hrs]

Open source Simulators: Different types of simulators for architectures, Classification parameters, Different modes of simulators, Building benchmarks and running sample codes to observe different parameters, Case Study: Any one of the simulator like gem5 or Tejas.

[7 Hrs]

Domain Specific-Architectures: Guidelines for domain specific architectures – Deep Learning Architecture - Google's Tensor Processing Unit (TPU) for Deep Neural Networks (DNNs) - Pixel Visual Core, a Personal Mobile Device Image Processing Unit, Secured architecture practices and approaches.

[6 Hrs]

Text Books

- Gerassimos Barlas, "Multicore and GPU Programming: An Integrated Approach", Morgan Kaufmann, 2015, ISBN: 978-0-12-417137-4.
- Rob Oshana, "Multicore Application Development Techniques: Applications, Tips and Tricks", Elsevier, 2016, ISBN: 978-0-12-800958-1.
- John L Hennessy, David A Patterson, "Computer Architecture: A Quantitative Approach", Sixth Edition, Morgan Kaufmann, 2018, ISBN: 978-0-12-811905-1.

Reference Books

- Rajeev Balasubramonian, Norman P. Jouppi, and Naveen Muralimanohar, "Multicore Cache Hierarchies", Morgan & Claypool Publishers, 2011,ISBN: 9781598297546.
- Daniel J. Sorin, Mark D. Hill, David A. Wood "A Primer on Memory Consistency and Cache Coherence", Morgan & Claypool Publishers, 2011, ISBN: 9781608455652.

(CT(DE)-22037) Multicore Technology Laboratory

Teaching Scheme: **Examination Scheme:**

Laboratory : 2 Hours/week	Continuous evaluation: 30 Marks Mini Project/ Assignments: 20 marks End Semester Exam: 50 Marks

Course Outcomes:

Students will be able to:

1. Explore programming models for multicore systems.
2. Program multicore systems using OpenMP.
3. Analyze differences in different simulator approaches.
4. Anaylzing outcomes of architectural simulations.
5. Memory behavior analysis using different benchmarks.

List of Assignments:

1. Program for matrix vector multiplication using pthreads.
2. Program to implement data sharing and thread scheduling.
3. Program for matrix vector multiplication using OpenMP clauses.
4. Program to implement sorting using OpenMP clauses.
5. Program to implement function parallel aspect using OpenMP clauses.
6. Program to demonstrate the use of barriers, critical sections.
7. Program to implement ordered and reduction clauses.
8. Setting up a open source simulator.
9. Running basic programs with open source simulator.
10. Benchmark study and analysis.

Reference Books

- Barbara Chapman, Gabriele Jost, Ruud van der Pas, "Using OpenMP Portable Shared Memory Parallel Programming", The MIT Press, 2008, ISBN-13: 978-0-262-53302-7.
- Online resources for simulators

(CT(DE)-22032) Geographical Information Systems

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. comprehend fundamental concepts and practices of Geographic Information Systems (GIS) and advances in Geospatial Information Science and Technology (GIS&T).
2. have a basic, theoretical and practical understanding of GIS.

3. work independently with various types of geographical data in GIS.
4. understand maps, images and apps to communicate spatial data in a meaningful way.
5. apply GIS analysis to address geospatial problems and/or research questions.

Course Contents

Introduction: Introduction to GIS and Digital Geographic Data & Maps Introduction to Digital Geographic Data: Introduction to Geographic Information Systems, Spatial Measurement, Spatial Location and Reference, Spatial Patterns, Geographic Data Collection Map Basics: Abstract Nature of Maps, Map Scale, More Map Characteristics, Map Projection, Grid Systems for Process, Map Symbolism, GIS Data Models.

[6 Hrs]

Input, Storage and Editing: The Input Subsystem: Primary Data, Input Devices, Vector Input, Raster Input, Remote Sensing Data Input, GPS Data Input, Metadata and Metadata Standards. Data Storage and Editing: Storage of GIS Databases, Detecting and Editing Errors of Different Types, Dealing with Projection Changes, Edge Matching, Rubber Shitting.

[8 Hrs]

Analysis: Elementary Spatial Analysis: GIS Data Query, Defining Spatial Characteristics, Working with Higher – Level Objectives Measurement: Measuring Length of Linear Objectives, Polygons, Shape and Distance Classification: Classification Principal, Elements of Reclassification, Neighborhood Functions, Roving Windows, Buffers Statistical Surfaces: Surface Mapping, Sampling the Statistical Surface, The DEM, Raster Surface, Interpolation, Terrain Reclassification, Slicing the Statistical Surface, Cut and Fill Spatial Arrangement Point, Line and Area Arrangement, Point Patterns, Thiessen Polygons, Area Patterns, Distance and Adjacency,Polygon Arrangement Measures, Linear Patterns, Directionality of Linear and Areal Objective, Connectivity of Linear Objects, Gravity Model, Routing and Allocation

[10 Hrs]

The Cartographic Overlay, Point-in-Polygon, Line-in-Polygon, Polygon Overlay, Automating the Overlay, Types of Vector Overlay, CAD-Type Overlay, Dasymetric Mapping Cartographic Modeling: Model Components, The Cartographic Models, Types of Cartographic Models, Inductive and Deductive Modeling, Factor Selection, model Flowchart, Model implementation, Model Verification

[8Hrs]

GIS Output: The Output from Analysis: Output: The Display of Analysis, Cartographic Output, The Design Process, Map Design Controls, Noncartographic Output, Two Case Studies on GIS, GIS application areas :Urban management, Land resources, Environment, Transportation

[8Hrs]

Text Books

- Michael N DeMers, “Fundamentals of Geographic Information Systems”, Wiley India Education
- Kang-tsung Chang, “Introduction to Geographic Information Systems”, McGraw-Hill Publication

Reference Books

- YEUNG, ALBERT K. W., LO, C. P., “Concepts and Techniques of Geographic Information Systems”, PHI Learning
- Victor Mesev, “Integration of GIS and Remote Sensing”, Baker-Berry G70.212 .I573 2007, ISBN: 0470864109

(CT(DE)-22038) Geographical Information Systems Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. become proficient in the use of GIS tools to conduct spatial analyses and build maps that are fit-for-purpose and effectively convey the information.
2. become effective in building maps that can be shared with non-GIS users (e.g. PDF maps and interactive webGIS maps)
3. describe pre-processing requirements and discuss various Digital Image Processing techniques.
4. rationalise statistical outlook of satellite images and different classification approaches with respect to diverse applications
5. elucidate integrated geospatial techniques and apply them in solving real world problems.

Suggested List of Assignments

1. Study of Basics of any open source GIS Tool (eg. Qgis)
2. Coordinate Systems and Map Projections in GIS Tool.
3. Working with Tables, Queries, and Basic Geoprocessing Tools
4. Creating and Editing new spatial data files in GIS Tool
5. Semantic Segmentation
6. Digital Elevation Model analysis
7. Implementation of Mini project

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-22039) Introduction to Blockchain, Cryptocurrencies, and Smart Contracts

Teaching Scheme	**Examination Scheme**
Lectures: 2 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Discuss the basics of Blockchain and cryptography.
2. Discuss the internal working of various cryptocurrencies.
3. Analyze different Blockchain platforms and smart contracts.
4. Write a working application on any of the Blockchain platform.
5. Demonstrate the implementation of smart contracts.

Course Contents

Introduction to Blockchain: History, need of Blockchain, Blockchain basics, changing landscape of digitization, cryptographic concepts, hashing, public and private Blockchain, smart contracts and consensus mechanisms.

[8 Hrs]

Introduction to Cryptocurrencies: Internal working of crypto currencies like Bitcoin, Ethereum, Litecoin, Dodgecoin, altcoin, etc.

[8 Hrs]

Blockchain Platforms: Hyperledger Fabric, Sawtooth, Multichain, R3 Corda, Steller, Ripple, Cloud based Blockchain, Comparison of various Blockchain platforms, Writing smart contracts in Ethereum and Hyperledger Fabric.

[8 Hrs]

Blockchain Use Cases: Blockchain applications in Finance, Banking, Retail, Land Records, Loyalty Programs, Agriculture, and Supply Chain, implementation of any of the use case on the chosen Blockchain platform.

[8 Hrs]

Emerging Trends in Blockchain: DeFi, utility tokens, integration of Blockchain with other technologies like IoT, AI, ML, Data Science, etc., Blockchain as a service, privacy and security issues in Blockchain.

[8 Hrs]

Text Books

- "Blockchain for Dummies", Tiana Laurence, Second Edition, 2017, Wiley Publication, ISBN 978-1119555018

- "Blockchain Quick Reference: A guide to exploring decentralized blockchain application development", Brenn Hill, Second Edition, 2015, Pearson Education India; ISBN 81-203-0596-5

Reference Books

- "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction", Arvind Narayanan, 2016, Princeton University Press
- "Mastering Blockchain", Imran Bashir, 2018, Packt Publication, ISBN 978-1788839044

(CT(DE)-22041) Introduction of Blockchain, Cryptocurrencies, and Smart Contracts Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs/ Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Demonstrate Blockchain Node and network setup on Cloud platforms.
2. Implement client side applications on Blockchain SDK.
3. Understand and implement the cryptocurrencies transactions on Blockchain platforms like Ethereum.
4. Write a working end-to-end application on multiple Blockchain platforms.
5. Demonstrate the implementation of smart contracts.

Assignments:

1. Create a simple Blockchain node setup in any cloud environment (AWS, Azure, GCP, etc.)
2. Implement a Blockchain network on the Multichain platform
3. Use Geth to Implement Private Ethereum Blockchain.
4. Write a Blockchain application to transfer the ether cryptocurrency from one user to another.
5. Build Hyperledger Fabric client application for a supply chain application.
6. Build Hyperledger Fabric application and write a smart contract to transfer ownership of assets in a supply chain.
7. Create a case study of Blockchain being used in illegal activities in real world.
8. Build an application on Corda platform for the student loan disbursal.

Above is a suggested list of assignments. The instructor is expected to continuously improve it.

(CT(DE)-22040) GPU Computing

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Learn and comprehend the differences in CPU & GPU.
2. Demonstrate the need of GPU computing.
3. Analyze different possible performance issues associated with GPU.
4. Program massively parallel processors with different ways.
5. Apply the practices for different problems.

Course Contents

Introduction to GPU: History, graphics processors, graphics processing units, GPGPUs. Clock speeds, CPU / GPU comparisons, heterogeneity, Accelerators, Types of cores, Latency hiding architectures, SIMT programming model, Compute Levels, Different microarchitectures, Basics of GPU architectures.

[5 Hrs]

CUDA Architecture: CUDA Architecture overview, Setting-up C based CUDA environment, General flow of a simple CUDA program & discussions, Grids, Blocks, Threads, Multi-GPU solutions & Streams, Parallel Patterns, Compiling, Debugging & Thread management, Optimizing CUDA applications with different considerations with case studies.

[8 Hrs]

GPU Memory: Unified memory, Global memory, Shared memory, Cache structure, Constant ,memory & events, Texure memory, Examples of all memory types, Memory handling with CUDA, Memory consistency. Barriers (local versus global), atomics, memory fence. Prefix sum, reduction. Programs for concurrent data structures.

[7 Hrs]

Programming GPU with Python based approaches: pyCUDA, Simple examples using PyCUDA syntax, Differences in C based CUDA & pyCUDA approaches, Heterogeneous programming with pyCUDA, Memory management with pyCUDA, An overview of pyOpenCL approaches & differences.

[6 Hrs]

OpenACC: Parallel programming with OpenACC, Basics of OpenACC, Loop level parallelism, Programming tools, Best practices for OpenAcc program development, OpenACC & performance portability, Interoperability.

[7 Hrs]

Applications in different doamins: Additional approachs to parallel programming, Docker-Container, Process and workflow of docker-container usages, Different environments- Tensorflow & Pytorch, Deep Learning Accelerations with CUDA, Examples of different domain datasets & Performance observations.

[7 Hrs]

Text Books

- David Kirk, Wen-mei Hwu, "Programming Massively Parallel Processors: A Hands-on Approach", Morgan Kaufman, 2010, ISBN: 978-0123814722
- Sunita Chandrasekaran, Guido Juckeland, "OpenACC for Programmers Concepts and Strategies", Addison-Wesley, 2018, ISBN-13: 978-0-13-469428-3
- Giancarlo Zaccone, "Python Parallel Programming Cookbook", Second Edition, Packt Publishing, 2019, ISBN 978-1-78953-373-6

Reference Books

- Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufman, 2012, ISBN: 978-0124159334
  Jaegeun Han, Bharatkumar Sharma"Learn CUDA Programming", Packt Publishing, 2019, ISBN 978-1-78899-624-2

(CT(DE)-22042) GPU Computing Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks
	End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Program GPU with different constructs.
2. Compile, debug the code & apply optimizations for performance observations.
3. Analyze different possible solutions to a program and select the most efficient one.
4. Apply different approaches to implement parallel programming .
5. Demonstrate the need of all studied approaches.

Suggested List of Assignments

1. Set-up the CUDA environment in Linux & Windows. Finding out the microarchitecture & compute capability of the available graphics card/ device.
2. A simple CUDA program and discussions to all complonents.
3. A program to implement vector addition to understand the grid, block & thread components.
4. A program to implement shared memory with multiple block and multiple threads.
5. A program to implement constant memory & events with example like ray tracing.
6. A program to implement texture memory with example like heat transfer simulation.
7. A simple program to understand pyCUDA environment with example like prefix sum.

8. A program to implement a sorting algorithm using pyCUDA syntax.
9. A program to observe the OpenACC flow and simple clauses.
10. A program to implement loop level parallelism using OpenACC
11. A program to understand the working flow of Docker-container concept.

This is a suggested list. The instructor is expected to continuously update it.

Reference Books

- Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufman, 2012, ISBN: 978-0124159334
- Jaegeun Han, Bharatkumar Sharma"Learn CUDA Programming", Packt Publishing, 2019, ISBN 978-1-78899-624-2

MINOR in Computer Engineering

(CT(MI)-22001) Internet Technologies (Minor)

Teaching Scheme:	**Examination Scheme:**
Lectures : 3 Hrs/week	Assignment/Quizzes – 40 marks
	End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Describe, draw diagrams, solve analytical problems based on, structure of computer networks.
2. Describe, solve analytical problems based on, networking protocols
3. Create WWW pages to serve as front-end for Internet applications
4. Demonstrate the ability to write server side and client side programs.

Course Contents

Unit I: Introduction: Introduction to Internet, Evolution of Internet, Internet applications, Protocols and Standard, Important Internet Services: WWW, File Transfer, Email, DNS, Remote Access: Telnet, SSH, Search Engines, Browser Basics

[6 Hrs]

Unit II: Computer Network – Overview: Introduction to computer network, OSI Model: Layered Architecture, Functions of the layers, Peer-to-Peer Processes, Encapsulation, TCP/IP protocol suite: Addressing, Packet forwarding, Introduction to UDP.

[8 Hrs]

Unit III: Web Essentials: Clients, Servers, Communication, HTTP Request Message, HTTP Response Message, Web Clients, Generations of web applications.

[6 Hrs]

Unit IV : Markup languages : An Introduction to HTML, Fundamental HTML Elements head, body etc. Document publishing, Introduction to XML. Introduction to Cascading Style Sheets, CSS features, CSS syntax, Style properties of text, box, layout, list, table, cursor etc

[6 Hrs]

Unit V : Client-Side Programming : Introduction to JavaScript, Basic Syntax, Variables and Data Types, Statements, Operators, literals, functions. Javascript Objects – properties, references, methods, constructors. Arrays, other built-in objects. Debugging javascript. Browsers.

[6 Hrs]

Unit VI : Server-Side Programming : PHP - Client Request – form data, request headers. Server Response - HTTP Status Codes, HTTP Response Headers. Sessions, Cookies, URL Rewriting. Introduction to Web services.

[8 Hrs]

Text Books

- Jeffrey C.Jackson, "Web Technologies : A Computer Science Perspective", Pearson Education, 2nd edition, 2007
- B. A. Forouzan and Firouz Mosharraf, Computer Networks, A Top-Down Approach, Tata McGraw-Hill, 2012

Reference Books

- A S Tanenbaum, "Computer Networks", 4th Edition, Pearson Education, ISBN 9788177581652
- Marty Hall, Larry Brown, "Core Web Programming", Pearson Education, 2nd Edition, 2001.
- Robert. W. Sebesta, "Programming the World Wide Web", Pearson Education, 4th Edition, 2007.
- H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web: How To Program", Pearson Education, 3rd Edition, 2006.

(CT(MI)-22002) Data Science (Minor)

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Design data understanding model.
2. Build data preprocessing model.
3. Analyze the visualized data.
4. Apply data model to available data.
5. Analyze performance of data model.

Course Contents

Data Understanding: Introduction to Data Science, Applications, Sources of data, Data types, Using case study - Describing data, their structure, their relevance, their records type.

[6 Hrs]

Data Preprocessing: Introduction to data preprocessing, Data features, Statistical data types, Data preparation - data cleaning and integration, data transformation, ETL model. Data Preprocessing with Pandas- DataFrame data structure, indexing and loading, querying a dataframe, Missing values, Merging, Groupby, pivot table, date/time functionality.

[10 Hrs]

Exploratory Data Analysis: Descriptive and inferential statistics, Chart types- Single var: Dot plot, Jitter plot, Error bar plot , Box-and-whisker plot, Histogram. Two variable: Bar chart, Scatter plot, Line plot, Log-log plot, Basic Charting using Matplotlib and Seaborn.

[10 Hrs]

Data Modeling: Data representation, Building a hypothesis model, Cost function, Linear regression, Logistic regression, Classification, Hypothesis evaluation, Model selection, Dataset introduction – training, testing, validation.

[8 Hrs]

Model Evaluation: Confusion matrix, precision, recall, f-measure, accuracy, sensitivity, specificity. Case studies: Weather forecast data, Loan prediction data, Sports data.

[4 Hrs]

Text Books

- "Mining of Massive Datasets", Jure Leskovec, Anand Rajaraman, and Jeffery David Ullman, Cambridge University Press, 2 edition (13 November 2014) , ISBN-10: 1107077230, ISBN-13: 978-1107077232
- “Foundations of Data Science”, Avrim Blum, John Hopcroft, and Ravindran Kannan, Hindustan Book Agency, (online free version) January 2020, ISBN-10: 9386279800

Reference Books

- "Machine Learning", Tom Mitchell, McGraw-Hill, 1st Ed May 2013, ISBN-10: 1259096955| ISBN-13: 978-1259096952.
- Pang-Ning Tan, Michael Steinbach, Vipin Kumar, “Introduction to Data Mining”, First Editin, Pearson Education, ISBN-13: 978-0321321367
- Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 1st Edition by Wes McKinney , O’Reilly

HONOURS in Computer Engineering

(CT(HO)-22001) Deep Learning (Honours)

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 Hrs/week	Assignment/Quizzes – 40 marks
	End Sem Exam - 60 marks

Prerequisites:

Linear Algebra, Vector Calculus, Probability Theory, Machine Learning

Course Outcomes:

Students will be able to:

1. Demonstrate the knowledge of the fundamentals of neural networks.
2. Design feedforward networks with backpropagation.
3. Develop deep learning model addressing real-world problems.
4. Evaluate the performance of neural network models.
5. Analyze the performance and optimize the deep learning model by undertaking a case study.

Basics: Linear Algebra Primer, Vector Calculus Review, Brief review of concepts from Linear Algebra and Vector Calculus.

[2 Hrs]

Neural Networks: Biological Neural Network, Artificial Neural Network, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Convergence theorem for Perceptron Learning Algorithm, multilayer perceptrons (MLPs), feedforward neural networks, activation functions.

[4 Hrs]

Feedforward Networks and learning Algorithms: Single and multilayer perceptrons (MLPs), representation power of MLPs, sigmoid neurons, feedforward neural network representation, Gradient Descent, Delta rule, Backpropagation, learning weights from more than one node, backpropagating errors from more output nodes, backpropagating errors to more layers, working example of weight update.

[6 Hrs]

Optimization and Regularization: Types of errors, bias-variance trade-off, overfitting-underfitting, brief review of concepts from optimization, variants of gradient descent, momentum based methods (Gradient Descent, Batch Optimization, Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam), Saddle point problem in neural networks. Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Greedy Layerwise Pre-training, Better activation functions, weight initialization methods, Batch Normalization.

[8 Hrs]

Autoencoders: Unsupervised Learning with Deep Network, Autoencoders (AEs), Regularization in autoencoders, Types of autoencoders (Denoising, sparse, contractive)

[6 Hrs]

Convolutional Neural Networks (CNNs) and different CNN architectures: Building blocks of CNN, Transfer Learning, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet, Visualizing CNNs, Guided Backpropagation.

[8 Hrs]

Recurrent Neural Networks (RNN): RNN architecture, Backpropagation through time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, Long Short Term Memory network and variants of RNN (Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs), Encoder Decoder Models, Attention Mechanism, transformers

[8 Hrs]

Text Books

- Goodfellow, Y. Bengio, and A. Courville, "Deep learning" MIT press, 2016. ISBN 978-0262035613

Reference Books

- R. Rojas, "Neural networks: a systematic introduction". Springer Science and Business Media, 2013. ISBN:
- C. M. Bishop and N. M. Nasrabadi, "Pattern recognition and machine learning", vol. 4, no. 4. Springer, 2006. ISBN: 978-0387310732
- J. Krohn, G. Beyleveld, and A. Bassens, "Deep learning illustrated: a visual, interactive guide to artificial intelligence". Addison-Wesley Professional, 2019. ISBN: 978-0135116692

(CT(HO)-22003) Reinforcement Learning (Honours)

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Define the key features and tasks of RL.
2. Analyze the RL problem, formulate the problem in terms of the state space, action space, dynamics and reward model.
3. State the algorithm best suited for addressing the Rl problems.
4. Implement in code common algorithms following code standards and libraries used in RL.
5. Describe (list and define) multiple criteria for analyzing RL algorithms and evaluate algorithms

7.

Course Contents

Introduction to Reinforcement Learning: Machine learning paradigm, Elements of RL, Early History of RL, Applications, Limitations and scope. Finite Markov Decision Processes: The Agent–Environment Interface, Goals and Rewards, Returns and Episodes, Policies and Value Functions, Optimality and Approximation.

[6 Hrs]

Dynamic Programming: Policy Evaluation, Policy Improvement, Policy Iteration, Value Iteration, Asynchronous DP, Generalized Policy Iteration. Monte Carlo Methods: Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, on policy and off policy learning, Importance Sampling

[6 Hrs]

Temporal Difference learning: TD prediction, Optimality of TD(0), SARSA, Q-learning, Games and after states. Planning and Learning: Model based learning and planning, prioritized sweeping, Dyna, heuristic search, trajectory sampling, Planning at Decision Time, Rollout Algorithms, Monte Carlo Tree Search.

[8 Hrs]

Prediction with Approximation: Value-function Approximation, The Prediction Objective, Stochastic-gradient and Semi-gradient Methods, linear function approximation, ANN based function approximation, Least-Squares TD, On-policy Control with Approximation, Off-policy Methods with Approximation

[8 Hrs]

Policy Gradient methods: Policy approximation, Policy Gradient Theorem, REINFORCE algorithm, actor-critic methods, Policy Gradient for Continuing Problems, Policy Parameterization for Continuous Actions, Eligibility traces: n-step TD prediction, TD(lambda), forward and backward views, Q(lambda), SARSA(lambda), replacing traces and accumulating traces

[8 Hrs]

Applications and Case Studies: TD-Gammon, Samuel's Checkers Player, Watson's Daily-Double Wagering, Optimizing Memory Control, Human-level Video Game Play, Personalized Web Services, Thermal Soaring

[4 Hrs]

Text Books

- "Reinforcement Learning: An Introduction", R. S. Sutton and A. G. Barto, Second Edition, 2018, The MIT Press ,Cambridge, ISBN 978-0-262-19398-6

Reference Books

- "Deep Reinforcement Learning with Python", Nimish Sanghi, First Edition, 2021, Apress, ISBN 978-1-4842-6808-7

- "Deep Reinforcement Learning with Python", Sudharsan Ravichandiran, Second Edition, 2020, TPackt Publishing, ISBN 978-1-83921-068-6
- "Artificial Intelligence: A Modern Approach,Stuart Russell and Peter Norvig", Prentice Hall, 3rd Edition. ISBN 0-13-103805-2.

HONOURS in Information Security

(CT(HO)-22002) Malware Analysis

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Understand the different malware concepts and taxonomy, and progresses to hunting down and collecting malware samples, and finally how to analyze them effectively by using the right tools the right way
2. Mitigate threat and strengthen our defenses against future generations of similar malware attacks.
3. Helps to build better solutions that minimize the proliferation of malware and helps us detect and respond to compromises
4. Aware of the concepts of malware and the different tools available, a lot of patience and perseverance..

Course Contents

Malware Blueprint : Malware Analysis, Malware Analysis and Reverse Engineering, Types of Malware Analysis, Purpose of Malware Analysis, Limitations of Malware Analysis, The Malware Analysis Process, The Effective Malware Analyst, Familiarization with Malware and Analysis Tools

[8 Hrs]

ware Taxonomy and Deployment : Malware Classes, Malware Infection Vectors : Speed, Stealth, Coverage, and Shelf Life, Types of Malware Infection Vectors, Potential Infection Vectors

[10 Hrs]

Protective Mechanisms : The Two States of Malware : Static and Dynamic Malware, Protective Mechanisms: Static Malware Protective Mechanisms and Dynamic Malware Protective Mechanisms

[10 Hrs]

Malware Dependencies and Inspection : Dependency Types, Malware Inspection, The Portable Executable File, The Proper Way to Handle Files, Inspecting Static Malware, Inspecting Dynamic Malware, Malware Analysis Use Cases, Malware Analyst Toolbox

[12 Hrs]

Text Books

- “Advanced Malware Analysis”, Christopher C. Elisan, First Edition, 2015, McGraw Hill, ISBN 978-0071819749
- “Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware”, Abhijit Mohanta and Anoop Saldanha, First Edition, 2020, Apress, ISBN 978-1484261927

Reference Books

- “Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware”, Monnappa K A, First Edition, 2018, Packt Publishing Limited, ISBN 978-1788392501
- "Mastering Malware Analysis: The complete malware analyst's guide to combating malicious software, APT, cybercrime, and IoT attacks", Alexey Kleymenov and Amr Thabet, First Edition, 2019, Packt Publishing Limited, ISBN 978-1789610789.

(CT(HO)-22004) Internet of Things Security

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/ Week	Assignment/Quizzes : 40 marks
	End Semester Exam: 60 marks

Course Outcomes

Students will be able to:

1. Aware of the Security requirements in IoT.
2. Know the cryptographic fundamentals for IoT
3. Recognize the authentication credentials and access control
4. identify with the various types Trust models and Cloud Security.

Course Contents

Introduction to IoT Security Cybersecurity versus IoT security and cyber-physical systems, IoT uses today, The IoT in the enterprise The IoT of the future and the need to secure

[8 Hrs]

Security Engineering for IoT Primer on threats, vulnerability, and risks (TVR), Primer on attacks and countermeasures, Today's IoT attacks, Threat modeling an IoT system, Building security in to design and development, Secure design: Safety and security design, and Processes and agreements, Security products and services : IoT device hardware, Selecting an MCU, Selecting a real-time operating system (RTOS), IoT relationship platforms, Cryptographic security APIs, Authentication/authorization, Edge, and Security monitoring.

[8 Hrs]

IoT Security Lifecycle and Cryptographic Fundamentals The secure IoT system implementation lifecycle : Implementation and integration, Operations and maintenance, and Dispose, Cryptography and its role in securing the IoT, Cryptographic module principles, Cryptographic key management fundamentals, Examining cryptographic controls for IoT protocols, Future directions of the IoT and cryptography

[8 Hrs]

Identity, Access Management, and Privacy for IoT An introduction to identity and access management for the IoT, Authentication credentials, IoT IAM infrastructure, Authorization and access control, Privacy challenges introduced by the IoT, Guide to performing an IoT PIA, PbD principles, Privacy engineering recommendations

[8 Hrs]

Cloud Security for IoT Cloud services and the IoT, Exploring cloud service provider IoT offerings, Cloud IoT security controls, Tailoring an enterprise IoT cloud security architecture, New directions in cloud-enabled IOT computing

[8 Hrs]

Text Books

- "Internet of Things Security: Principles and Practice", Qinghao Tang Fan Du, First Edition, 2021, Springer, Singapore, ISBN 978-981-15-9942-2
- A Beginner’s Guide to Internet of Things Security Attacks, Applications, Authentication, and Fundamentals, Brij B. Gupta, Aakanksha Tewari, First Edition, 2020, Taylor & Francis, ISBN 9780367430696.

Reference Books:

- "Practical Internet of Things Security: Beat IoT security threats by strengthening your security strategy and posture against IoT vulnerabilities Kindle Edition", Brian Russell and Drew Van Duren, First Edition, 2016, Packt Publishing, ISBN : 978-1785889639
- "Practical Internet of Things Security: Design a security framework for an Internet connected ecosystem", Brian Russell and Drew Van Duren, 2nd Edition, 2018, Packt Publishing Limited, ISBN : 978-1788625821