Dr. Babasaheb Ambedkar Technological University, Lonere

(Established as a University of Technology in the State of Maharashtra)
(Under Maharashtra Act No. XXIX of 2014)
P.O. Lonere, Dist. Raigad, Pin- 402 103, Maharashtra
Telephone and Fax. : 02140 - 275142
www.dbatu.ac.in

Course Structure and Detailed Syllabus
for

Third Year
B. Tech. Programme in Information Technology
(Effective from Academic Year 2022-23)

Third Year B. Tech. Information Technology Syllabus (With effect from 2022-23)

	Semester V									
Course Category	**Course Code**	**Course Title**	**Teaching Scheme**			**Evaluation Scheme**				
			L	T	P	CA	MSE	ESE	Total	Credits
PCC	BTITC501*	Software Engineering	3	1	-	20	20	60	100	4
PCC	BTITC502	Computer Networks and Internetworking Protocols	3	1	-	20	20	60	100	4
PEC		**Elective- II**	3	-	-	20	20	60	100	3
	BTITPE503A	Embedded Systems								
	BTITPE503B	IT Service Management								
	BTITPE503C	Information Storage Management								
	BTITPE503D	Network Management								
	BTITPE503E	Data Visualization								
	BTITPE503F	Virtual Reality								
OEC		**Elective- III**	3	-	-	20	20	60	100	3
	BTITOE504A	Theory of Computation								
	BTITOE504B	Graph Theory								
	BTITOE504C	Programming in Java								
	BTITOE504D	Human Computer Interaction								
	BTITOE504E	Game Theory								
	BTITOE504F	3D Printing and Design								
LC	BTITL505	Computer Networks and Internetworking Protocols Lab	-	-	2	60	-	40	100	1
LC	BTITL506	Software Engineering and Elective- II Lab	-	-	4	60	-	40	100	2
Project	BTITP507	Mini Project - I	-	-	4	60	-	40	100	4
Internship	BTITF508	Internship – II Evaluation	-	-	-	-	-	-	-	Audit
			12	**2**	**10**	**260**	**80**	**360**	**700**	**21**
Semester VI										
PCC	BTITC601*	Operating Systems	3	1	-	20	20	60	100	4
PCC	BTITC602	Database Management Systems	3	1	-	20	20	60	100	4
PEC		**Elective- IV**	3	-	-	20	20	60	100	3
	BTITPE603A	Software Testing								
	BTITPE603B	Data Storage Technologies & Networks								
	BTITPE603C	Service Oriented Architecture								
	BTITPE603D	Network Programming								
	BTITPE603E	Data Warehousing & Data Mining								

OEC	BTITOE604A BTITOE604B BTITOE604C BTITOE604D BTITOE604E	**Elective- V** Compiler Design Enterprise Resource Planning Decision Support Systems Software Project Management Introduction to Data Science	3	-	-	20	20	60	100	3
LC	BTITL605	Database Management Systems Lab	-	-	2	60	-	40	100	1
LC	BTITL606	Operating Systems and Elective-IV Lab	-	-	4	60	-	40	100	2
Project	BTITP607	Mini Project - II	-	-	4	60	-	40	100	4
Internship	Internship- III	Field Training / Internship/Industrial Training (minimum of 4 weeks which can be completed partially in fifth semester and sixth semester or at one time).	-	-	-	-	-	-	-	To be audited in VII Sem.
			12	**2**	**10**	**260**	**80**	**360**	**700**	**21**

***** *These courses are to be studied on self –study mode using SWAYAM/NPTEL/Any other source.*

BSC = Basic Science Course, ESC = Engineering Science Course, PCC = Professional Core Course
PEC = Professional Elective Course, OEC = Open Elective Course, LC = Laboratory Course
HSSMC = Humanities and Social Science including Management Courses

Course Title:	**Software Engineering**	**Semester**	**V**
Course Code	**BTITC501**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**3 – 1 – 0**
Stream	**Core**	**Credits**	**4**

Course Objectives:

1. To understand software life cycle development models.
2. To understand and apply Software Requirements Engineering Techniques, Software design principles, modeling and Software testing techniques.
3. To understand the use of metrics in Software Engineering.
4. To understand Software Project Management.

Course Outcomes:

After learning the course, the students should be able:

1. To use the techniques, skills, and Modern Engineering tools necessary for Engineering practice.
2. To design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
3. To identify, formulate and solve Engineering Problems.

Course Contents:

UNIT I

Software Development Process: Software crisis and myths, Software process and development: Generic view of process, Software life cycle and models, Analysis and comparison of varies models, an agile view of process, Requirement Engineering: Requirements engineering tasks, Initiating requirement engineering process, Eliciting requirement, developing use-cases.

UNIT II

Building the analysis model, Negotiating and validating requirement, System Design Overview: Design process and design quality, Design concepts, Design model, Pattern based software design, Architectural design, User interface design. UML: Different methods: Rumbaugh / Booch / Jacobson, Need for standardization, Developing diagrams in UML (Use CASE, Class, Interaction, State diagrams).

UNIT III

Validation and Testing: Strategic approach to Software testing, Strategic issues, Test strategies for conventional software, Validation testing, System testing, Debugging, White box testing and Black box testing.

UNIT IV

Planning of Project: Project management, Metrics for process and projects, Estimation, Project scheduling.

UNIT V

Management of Project: Risk management, Importance of software quality and measurements, software engineering techniques for quality assurance, and Change management.

Text Books:

1. Roger S. Pressman, "***Software Engineering***", Tata McGraw-Hill, 7th Edition, 2018.
2. G. Booch, J. Rumbaugh, and I. Jacobson, "***The Unified Modeling Language User Guide***", Addison Wesley, 2nd Edition, 2005.

Reference Books:

1. Shari Pfleeger, "***Software Engineering***", Pearson Education, 4th Edition, 2012.
2. Ian Sommerville, "***Software Engineering***", Pearson Higher Education, 10th Edition, 2017.
3. Pankaj Jalote, "***An Integrated Approach to Software Engineering***", Springer New York, 2nd Edition, 2012.

Course Title:	**Computer Networks and Internetworking Protocols**	**Semester**	**V**
Course Code	**BTITC502**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**3 – 1 – 0**
Stream	**Core**	**Credits**	**4**

Course Objectives:

1. To understand the basic concepts of Computer Networks.
2. To Understand Network Layer and Applications.
3. To learn UDP and TCP applications.
4. To learn Transport Layer Reliability.

Course Outcomes:

After learning the course, the students should be able:

1. To compare and contrast TCP and UDP in terms of the application that uses them.
2. To design network-based applications using the socket mechanism.
3. To work with IPv4 addresses in terms of subnetting and supernetting.
4. To setup a host and network in terms of IP Addressing.
5. To trace the flow of a network packet over internet.
6. To design a network with subnets as specified.

Course Contents:

UNIT I

Introduction to Computer Networks and Underlying Technologies: Birth of the Internet, The Internet today, World Wide Web, Growth of the Internet, Protocols and Standards, Internet Standards, Internet Administration.

The OSI Model and the TCP/IP Protocol Suite:

Protocol Layers: Hierarchy Services, The OSI Model: Layered Architecture , Layer-to-Layer Communication, Encapsulation, Layers in the OSI Model, TCP/IP Protocol Suite: Comparison between OSI and TCP/IP Protocol Suite, Layers in the TCP/IP Protocol Suite, Addressing: Physical addresses, Logical addresses, Port addresses, Application-specific addresses, Wired Local Area Networks: IEEE Standard, Frame format, Addressing, Ethernet evolution, Standard Ethernet, Fast Ethernet, Gigabit Ethernet, Ten-Gigabit Ethernet.

UNIT II

Wireless LANS: IEEE standard, MAC Sublayer, Addressing Mechanism, Bluetooth, Point-to-Point WANs, Connecting devices: Repeaters, Bridges and Routers.

Introduction to Network Layer: Packet switching, Circuit switching, Packet switching at Network Layer, Network Layer Services, Other Network Layer Issues. IPv4 Addresses, Address Space Notation, Range of Addresses, Operations, Classful Addressing, Subnetting, Supernetting, Classless Addressing: Variable-Length Blocks, Two-Level Addressing, Block Allocation, Special Blocks, Special Addresses.

UNIT III

Delivery and Forwarding of IP Packets: Direct Delivery, Indirect Delivery, Forwarding, Structure of a Router, Components.
Internet Protocol Version 4(IPv4): Datagrams, Fragmentation, Maximum Transfer Unit (MTU), Fields Related to Fragmentation, Options: Format, Option Types, Checksum Calculation, IP Package.
Address Resolution Protocol (ARP): Address Mapping: Static Mapping, Dynamic Mapping, The ARP Protocol, ARP Package.
Internet Control Message Protocol (ICMP): Messages, Message Format, Error Reporting Messages, Query Messages, Checksum, Debugging Tools, ICMP Package.

UNIT IV

Introduction to Transport Layer: Transport-Layer Services, Process-to-Process communication, Addressing, Flow Control, Error Control, Congestion Control, Connectionless and Connection-Oriented Services.
User Datagram Protocol (UDP): User Datagram, UDP Services, Process-to-Process Communication, Connectionless Services, Flow Control, Error Control, UDP Applications, UDP Features, UDP Package.

UNIT V

Transmission Control Protocol (TCP): TCP Services, Process-to-Process Communication, Stream Delivery Service, Full-Duplex Communication, Multiplexing and Demultiplexing, Connection-Oriented Service, Reliable Service, TCP Features, Segment, Format, Encapsulation, A TCP Connection, State Transition Diagram, Scenarios, Windows in TCP, Flow Control, Silly Window Syndrome, Error Control, Data Transfer in TCP, Some Scenarios, Congestion Control, TCP Timers, TCP Package.

Text books:

1. Douglas E. Comer, "***Internetworking with TCP/IP: Principles, Protocols and Architecture***",Volume 1, 6th Edition, PHI publication, 2013.
2. Behrouz A. Forouzan, "***TCP-IP Protocol Suite***", 4th Edition, McGraw Hill publication, 2010.
3. A. S. Tanenbaum, "***Computer Networks***", 6th Edition, Pearson publication, 2022.

Reference books:

1. Comer, "***Internetworking with TCP-IP***", Volume 3, 6th Edition, Pearson publication, 2013.
2. W. Richard Stevens, "***UNIX Network Programming: Interprocess Communications***", Volume 2, 2nd Edition, PHI publication, 2002.
3. William Stalling, "***SNMP, SNMPv2, SNMPv3, and RMON 1 and 2***", 3rd Edition, Pearson education publication, 2006.
4. Hunt Craig, "***TCP-IP Network Administration***", 3rd Edition, O'Reilly publication, 2002.

Course Title:	**Embedded Systems**	**Semester**	**V**
Course Code	**BTITPE503A**	**Course Type**	**Elective**
Prerequisite	**Microprocessor & Microcontroller**	**L – T – P**	**3 – 0 – 0**
Stream	**Software Application and Development**	**Credits**	**3**

Course Objectives:

1. To understand the fundamental concepts in Embedded Systems.
2. To learn Real Time Operating Systems.
3. To get acquainted with hardware & interfaces.
4. To know Embedded System Design Techniques.

Course Outcomes:

After learning the course, the students should be able:

1. To demonstrate & explain embedded systems hardware & software components.
2. To define embedded systems using real time operating system – VxWorks/ μCOS II RTOS.
3. To design & develop embedded applications using C language.
4. To apply design techniques in real-life application.

Course Contents:

UNIT I

Introduction: Introduction to embedded systems overview, design challenges, common design metrics, processor technology, IC technology, Design technology. Design productivity gap.

ARM Architecture: ARM 7 processor fundamentals, memory management, ARM processor family, Instruction set & interfacing. Introduction to ASIPS, Microcontrollers and DSP.

UNIT II

Devices and Interfacing: Processor interfacing, Arbitration, Multilevel bus architecture. Basic protocol concepts: serial protocols, I2C,CAN, Firewire and USB, Parallel protocols, PCI bus, ARM bus, Wireless protocols: IrDA, Bluetooth, IEEE 802.11, Device Driver programming.

UNIT III

Programming concepts: State m/c & concurrent process model, FSM m/c, FSMD, PSM model & concurrent process model, Scheduling process, Data flow model, Embedding programming in C++, JAVA and program modeling concepts.

UNIT IV

Real Time OS: OS services, Process management, Memory management device, File &IO subsystem management, Interrupt routines in RTOS, RTOS task scheduling models, Securities issues, RTOS mCOS-II & RTOS VxWorks.

UNIT V

Design Examples and Case Studies: Personal Digital Assistants, Digital thermometer, Case Studies of digital camera, Smart card, Case study of coding for sending application layer byte stream on TCP/IP network using RTOS VxWorks.

Text Books:

1. Raj Kamal, ***"Embedded Systems Architecture, Programming and Design"***, McGraw Hill Publication, 3rd Edition, 2017.
2. Frank Vahid/Tony Givargis, ***"Embedded Systems Design"***, Wiley, 2002.
3. Andrew N. Sloss, ***"ARM System Developers Guide"***, ELSEVER Publication, 1st Edition, 2004.

Reference Books:

1. Wayne Wolf, ***"Computer as Components – Principles of Embedded Computing System Design"***, Morgan Kaufmann, 2005.
2. David E Simon, ***"An Embedded Software Primer"***, Addison Wesley Publication, 1st Edition, 2004.

Course Title:	**IT Service Management**	**Semester**	**V**
Course Code	**BTITPE503B**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3 – 0 – 0**
Stream	**Infrastructure & Security Management**	**Credits**	**3**

Course Objectives:

1. To introduce practical implementation of Information Technology Service Management (ITSM).
2. To understand how an integrated ITSM framework can be utilized to achieve IT business integration, cost reductions and increased productivity.
3. To learn the best practices of ITSM methodology.

Course Outcomes:

After learning the course, the students should be able:

1. To identify IT services as a means to provide functionality and value to customers.
2. To describe the needs and targets of the different stakeholders (service providers, customers, suppliers/partners) in the services value chain.
3. To demonstrate the value of a service management framework.
4. To explain the service management processes for given customers.
5. To select the appropriate tools to support a given designed service management solution.

Course Contents:

UNIT I

IT Infrastructure: Introduction, Challenges in IT Infrastructure Management, Design Issues of IT Organizations and IT Infrastructure, IT System Management Process, IT Service Management Process, Information System Design Process.

UNIT II

Service Delivery Process: Service Level Management, Financial Management, IT Service Continuity Management, Capacity Management & Availability Management.

Service Support Process: Configuration Management, Incident Management, Problem Management, Change Management & Release Management.

UNIT III

Storage Management: Storage, Backup, Archive and Retrieve, Disaster Recovery, Space Management, Database and Application Protection and Data Retention.

UNIT IV

Security Management: Computer Security, Internet Security, Physical Security, Identity Management, Access Control System and Intrusion Detection.

UNIT V

Case Studies on how IT Service Management and ITIL processes make IT efficient and save cost for organizations.

Text Book:

1. Phalguni Gupta, Surya Prakash and Umarani Jayaraman, ***"IT Infrastructure & Its Management"***, Tata McGraw-Hill Education, 2009.

Reference Books:

1. W. Ronald Hudson, Ralph C. G. Haas, Waheed Uddin, ***"Infrastructure Management: Integrating Design, Construction, Maintenance, Rehabilitation, and Renovation"***, McGraw-Hill, 1997.

2. Anita Sengar, ***"IT Infrastructure Management"***, S. K. Kataria and Sons, 2nd Edition, 2009.

Course Title:	**Information Storage Management**	**Semester**	**V**
Course Code	**BTITPE503C**	**Course Type**	**Elective**
Prerequisite	**Computer Architecture & Organization**	**L - T - P**	**3 – 0- 0**
Stream	**Information Management & Quality Control**	**Credits**	**3**

Course Objectives:

1. To evaluate storage architecture, understand logical and physical components of storage Infrastructure including storage subsystems.
2. To describe Storage Networking Technologies such as FC-SAN, NAS, IP-SAN and data archival solution–CAS.
3. To identify different Storage Virtualization Technologies and their benefits.
4. To understand and articulate business continuity solutions including backup, recovery technologies and local, remote replication solutions.
5. To define Information security and storage security domains and identify parameters of managing and monitoring Storage Infrastructure. Also describe common storage management activities and solutions.

Course Outcomes:

After learning the course, the students should be able:

1. To describe and apply storage technologies.
2. To identify leading Storage Technologies that provides cost-effective IT solutions for medium to large scale businesses and data centers.
3. To describe important Storage Technologies' features such as availability, replication, scalability and performance.
4. To design, analyze and manage clusters of resources.

Course Contents:

UNIT I

Introduction to Information Storage Management: Intelligent Storage System (ISS) and its components Implementation of ISS as high-end and midrange storage-arrays, Direct Attached Storage - Introduction to SCSI.

Introduction to parallel SCSI, SCSI Command Model - Storage Area Networks - Fiber Channel Connectivity, Login types, Topologies.

UNIT II

Storage Networking Technologies: Network Attached Storage- General purpose servers vs. NAS Devices - Benefits of NAS, NAS File I/O - NAS Components, Implementation, File Sharing protocols, I/O operations - IPSAN-ISCSI, Components of ISCSI- Content-Addressed Storage.

UNIT III

Storage Virtualization: Fixed Content and Archives, Types, Features, Benefits, CAS Architecture, object storage and Retrieval, examples - Storage Virtualization - forms of virtualization, SNIA Taxonomy - Storage virtualization configurations, challenges, Types of storage virtualization, Business

Continuity- Overview of emerging technologies such as Cloud storage, Virtual provisioning, Unified Storage, FCOE, FAST.

UNIT IV

Business Continuity and Recovery: Information Availability, BC Terminology, Life cycle, Failure analysis - Backup and Recovery- Backup purpose, considerations, Backup Granularity, Recovery considerations- Backup methods, process, backup and restore operations, Overview of emerging technologies - duplication, offsite backup.

UNIT V

Storage Security and Management: Storage security framework, Securing the Storage Infrastructure Risk triad - Managing the storage infrastructure, Monitoring the storage infrastructure, Identify key parameters and components to monitor in a storage infrastructure List key management activities and examples define storage management standards and initiative-Industry trend.

Text Book:

1. EMC Corporation, ***"Information Storage and Management"*** *by* Somasundaram Gnanasundaram, Alok Shrivastava, Wiley India, 2nd Edition, 2012.

Reference Books:

1. IBM, "***Introduction to Storage Area Networks and System networking"***, 5th edition, 2012.
2. Robert Spalding, "***Storage Networks: The Complete Reference"***, Tata McGraw Hill, Osborne, 6th reprint 2003.
3. Marc Farley, "***Building Storage Networks"***, Tata McGraw Hill, Osborne, 1st Edition, 2001.
4. Tom Clark, "***Designing Storage Area Networks -A Practical Reference for Implementing Fiber Channel and IP SANs",*** Tata McGraw Hill 2003, 2nd edition.

Course Title:	**Network Management**	**Semester**	**V**
Course Code	**BTITPE503D**	**Course Type**	**Elective**
Prerequisite	**Computer Networks & Internetworking Protocols**	**L – T – P**	**3 – 0– 0**
Stream	**Network**	**Credits**	**3**

Course Objectives:

1. To understand the principles of network management, different standards and protocols used in managing complex networks.
2. To understand the automation of network management operations and making use of readily available network management systems.

Course Outcomes:

After learning the course, the students should be able:

1. To acquire the knowledge about network management standards (OSI and TCP/IP).
2. To acquire the knowledge about various network management tools and the skill to use them in monitoring a network.
3. To analyze the challenges faced by Network Managers.
4. To evaluate various commercial Network Management Systems and Open Network Management Systems.
5. To analyze and interpret the data provided by an NMS and take suitable actions.

Course Contents:

UNIT I

Data communication and network management overview: Analogy of Telephone Network Management, Communications protocols and Standards, Case Histories of Networking and Management, Challenges of Information Technology Managers, Network Management: Goals, Organization, and Functions, Network and System Management, Network Management System Platform, Current Status and future of Network Management.

SNMPV1 Network Management Organization and Information Models, Managed network, Managed network: Case Histories and Examples.

UNIT II

The History of SNMP Management, The SNMP Model, The Organization Model, System Overview, The Information Model.

SNMPV1 Network Management Communication and Functional Models: The SNMP Communication Model, Functional model. SNMP MANAGEMENT SNMPv2: Major Changes in SNMPv2, SNMPv2 System architecture, SNMPv2 Structure of Management Information, The SNMPv2 Management Information Base, SNMPv2 Protocol, Compatibility with SNMPv1.

SNMP MANAGEMENT RMON: What is Remote Monitoring? RMON SMI and MIB, RMON1, RMON2, ATM Remote Monitoring, Case Study of Internet Traffic Using RMON.

UNIT III

Telecommunication Management Network: Why TMN? Operations Systems, TMN Conceptual Model, TMN Standards, TMN Architecture, TMN Management Service Architecture, TMN Integrated View, Implementation.

UNIT IV

Network management tools and systems: Network Management Tools, Network Statistics Measurement Systems, Network Management systems, Commercial Network Management Systems.

UNIT V

Web-Based Management: NMS with Web Interface and Web-Based Management, Web Interface to SNMP Management, Embedded Web-Based Management, Desktop management Interface, Web-Based Enterprise Management, WBEM: Windows Management Instrumentation, Java management Extensions, Management of a Storage Area Network, Future Directions. Case Studies.

Text Book:

1. Mani Subrahmanian, ***"Network Management Principles and Practice"***, Pearson Education, 2nd Edition, 2010.

Reference Books:

1. Morris, ***"Network Management"***, Pearson Education, 1st Edition, 2008.
2. Mark Burges, ***"Principles of Network System Administration"***, Wiley DreamTech, 1st Edition, 2008.

Course Title:	**Data Visualization**	**Semester**	**V**
Course Code	**BTITPE503E**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3 – 0– 0**
Stream	**Data Science**	**Credits**	**3**

Course Objectives:

1. To understand various steps in data visualization.
2. To read data from different file formats such as .csv, .xlsx, JSON, .txt etc. in to the data frames.
3. To learn the nature of data and its domains and the concepts and skills of data visualization by understanding, questioning, and problematizing how data are generated, analysed, and used.
4. To learn time series data and its visualization.
5. To understand basic and advance chart types used in data visualization.

Course Outcomes:

After learning the course, the student will be able:

1. To list out various stages of the data visualization.
2. To identify/choose suitable data for the specific data visualization problem.
3. To plot useful plots/charts for data visualization problem under consideration.
4. To interpret the finding from different types of charts/graphs.
5. To select the right graph/chart to review datasets.

Course Contents:

UNIT I

The seven stages of Data Visualization: Why data display requires planning, example, Iteration and Combination, Principles. Getting Started with Processing: Sketching with processing, Example and Distributing your work, Examples and references, Functions, Sketching and Scripting
Mapping: Drawing a Map, Locations on map, Data on Map, Using your own data, Next step.

UNIT II

Types of Digital Data: Data abstraction, Store, Classification of Digital Data, Structured versus Unstructured Data. Reading Data from Varied Data Sources into Python Data Frame: Read from Excel Data Source, Read Data from .csv, load a Python Dictionary into a Data Frame, Reading JSON data into a Pandas Data Frame, Reading Data from Microsoft Access Database, Reading Data from .txt File, Reading Data from XML File.

UNIT III

Pros and Cons of Charts: Pie Chart, Tree Map, Heat Map, Scatter Plot, Histogram, Word Cloud, Box Plot, chart chooser. Good Chart Designs: Mistakes That Can Be Avoided, Less Is More, Tables versus Charts. Animated bar char race, interactivity in plots.

UNIT IV

Data Wrangling in Python: Pandas Data Manipulation, Dealing with Missing Values, Date Reshaping, Filtering Data Merging Data, Subsetting Data Frames in Pandas, Reshaping the Data and Pivot Tables Backfill, Forward Fill. Functions in Python Pandas: Pandas DataFrame Functions, Pandas Correlations, Pandas DataFrame All Method. Matplotlib for Data Visualization: Exploratory Data Analysis using Python, Matplotlib.

UNIT V

Plotly for Data Visualization: Plotly Python Package, Seaborn for Data Visualization, Seaborn Plots Using “iris” Dataset, Seaborn Plots Using “Superstore” Dataset, Seaborn Plots Using “OLYMPIC” Dataset, Seaborn Plots Using “Passengers Flights” Dataset. Time series and spatial data visualization, stock market data visualization.

Text Book:

1. Seema Acharya***,”Reimagining Data Visualization using Python”,*** Wiley Publication, 2022.
2. Ben Fry, ***“Visualizing Data: Exploring and Explaining data with Processing Environment”***, Shroff/O’Reilly Media, 2016.

Reference Books:

1. Scott Murray, ***“Interactive Data Visualization for the web”***, Shroff/O’Reilly Media, 2016.
2. Tamara Munzner, “Visualization Analysis and Design”, CRC Press, 2014.
3. Julia Steele, Noah Lliinsky, ***“Designing Data Visualizations”***, Shroff/O’Reilly Media, 2012.
4. Kyran Dale, ***“Data Visualization with Python and JavaScript: Scrape, Clean, Explore &Transform your data”***, Shroff/O’Reilly Media, 2016.
5. Julia Steele, Noah Lliinsky, ***“Beautiful Visualization”***, Shroff/O’Reilly Media, 2016.

Course Title:	**Virtual Reality**	**Semester**	**V**
Course Code	**BTITPE503F**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3 – 0– 0**
Stream	**---**	**Credits**	**3**

Course Objective:

1. To provide a detailed understanding of the concepts of Virtual Reality.

Course Outcomes:

After learning the course, the student will be able:

1. To understand geometric modeling and Virtual environment.
2. To study about Virtual Hardware and Software.
3. To develop Virtual Reality applications.

Course Contents:

UNIT I

Introduction to Virtual Reality: Virtual Reality and Virtual Environment: Introduction, Computer graphics, Real time computer graphics, Flight Simulation, Virtual environment requirement, benefits of virtual reality, Historical development of VR, Scientific Landmark.

3D Computer Graphics: Introduction, The Virtual world space, positioning the virtual observer, the perspective projection, human vision, stereo perspective projection, 3D clipping, Colour theory, Simple 3D modelling, Illumination models, Reflection models, Shading algorithms, Radiosity, Hidden Surface Removal, Realism-Stereographic image.

UNIT II

Geometric Modelling: Geometric Modelling: Introduction, From 2D to 3D, 3D space curves, 3D boundary representation, Geometrical Transformations: Introduction, Frames of reference, Modelling transformations, Instances, Picking, Flying, Scaling the VE, Collision detection Generic VR system: Introduction, Virtual environment, Computer environment, VR technology, Model of interaction, VR Systems.

UNIT III

Virtual Environment: Animating the Virtual Environment: Introduction, The dynamics of numbers, Linear and Nonlinear interpolation, the animation of objects, linear and non-linear translation, shape & object inbetweening, free from deformation, particle system.

Physical Simulation: Introduction, Objects falling in a gravitational field, Rotating wheels, Elastic collisions, projectiles, simple pendulum, springs, Flight dynamics of an aircraft.

UNIT IV

VR Hardware and Software: Human factors: Introduction, the eye, the ear, the somatic senses. VR Hardware: Introduction, sensor hardware, Head-coupled displays, Acoustic hardware, Integrated VR systems. VR Software: Introduction, Modelling virtual world, Physical simulation, VR toolkits, Introduction to VRML.

UNIT V

VR Applications: Introduction, Engineering, Entertainment, Science, Training. The Future: Virtual environment, modes of interaction.

Text Books:

1. John Vince, "***Virtual Reality Systems***", Pearson Education Asia, 2007.
2. Anand R., "***Augmented and Virtual Reality***", Khanna Publishing House, Delhi.
3. Adams, "***Visualizations of Virtual Reality***", Tata McGraw Hill, 2000.
4. Grigore C. Burdea, Philippe Coiffet ,"***Virtual Reality Technology***", Wiley Inter Science,2nd Edition, 2006.
5. William R. Sherman, Alan B. Craig, "***Understanding Virtual Reality: Interface, Application and Design***", Morgan Kaufmann, 2008.

Course Title:	**Theory of Computation**	**Semester**	**V**
Course Code	**BTITOE504A**	**Course Type**	**Elective**
Prerequisite	**Discrete Mathematics, Data Structure & Applications**	**L – T – P**	**3 – 0– 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To understand problem classification and problem solving by machines.
2. To understand the basics of automata theory and its operations.
3. To study and compare different types of computational models.
4. To encourage students to study theory of computability and complexity.
5. To understand the P and NP class problems and its classification.
6. To understand the fundamentals of problem decidability and reducibility.

Course Outcomes:

After learning the course, students should be able:

1. To construct finite state machines to solve problems in computing.
2. To write mathematical expressions for the formal languages.
3. To apply well defined rules for syntax verification.
4. To construct and analyse Push down Automata and Turing Machine for formal languages.
5. To express the understanding of the decidability and decidability problems.
6. To express the understanding of computational complexity.

Course Contents:

UNIT I

Finite State Machines and Automata theory: Symbols, Strings, Language, Formal Language, Natural Language, Basic Machine and Finite State Machine. Definition and Construction-DFA, NFA, NFA with epsilon-Moves, Minimization Of FA, Equivalence of NFA and DFA, Conversion of NFA with epsilon moves to NFA, Conversion of NFA with epsilon moves to DFA, FSM with output.

UNIT II

Regular Expressions: Definition and Identities of Regular Expressions, Construction of Regular Expression of the given L, Construction of Language from the RE, Construction of FA from the given RE using direct method, Conversion of FA to RE using Arden's Theorem, Pumping Lemma for RL, Closure properties of RLs, Applications of Regular Expressions..

UNIT III

Context free Grammar: Introduction, Formal Definition of Grammar, Notations, Derivation Process: Leftmost Derivation, Rightmost Derivation, derivation trees, Context Free Languages, Ambiguous CFG, Removal of ambiguity, Simplification of CFG, Normal Forms, Chomsky Hierarchy, Regular grammar, equivalence of RG (LRG and RLG) and FA.

UNIT IV

Push down Automata: Introduction and Definition of PDA, Construction (Pictorial/ Transition diagram) of PDA, Instantaneous Description and Acceptance of CFL by empty stack and final state, Deterministic PDA Vs Nondeterministic PDA, Closure properties of CFLs, pumping lemma.

UNIT V

Turing Machine: Formal definition of a Turing machine, Recursive Languages and Recursively Enumerable Languages, Design of Turing machines, Variants of Turing Machines: Multi-tape Turing machines, Universal Turing Machine, Nondeterministic Turing machines. Comparisons of all automata. Undecidability and Computational Complexity.

Text Books:

1. Michael Sipser, "***Introduction to the Theory of Computation",*** CENGAGE Learning, 3rd Edition, 2014.
2. Vivek Kulkarni, ***"Theory of Computation",*** Oxford University Press, 2013.

Reference Books:

1. Hopcroft Ulman, "***Introduction to Automata Theory"***, Languages and Computations, Pearson Education Asia, 3rd Edition, 2006.
2. DanielI A. Cohen, ***"Introduction to Computer Theory"***, Wiley-India, 2nd Edition, 1996.
3. K.L.P Mishra, N. Chandrasekaran, ***"Theory of Computer Science (Automata, Languages and Computation)"***, Prentice Hall India, 3rd Edition, 2006.

Course Title:	**Graph Theory**	**Semester**	**V**
Course Code	**BTITOE504B**	**Course Type**	**Elective**
Prerequisite	**Discrete Structures and Applications**	**L – T – P**	**3 – 0 – 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To understand and apply the fundamental concepts in graph theory.
2. To apply graph theory based tools in solving practical problems.
3. To improve the proof writing skills.

Course Outcomes:

After learning the course, the students should be able:

1. To solve problems using basic graph theory.
2. To identify induced sub graphs, cliques, matchings, covers in graphs.
3. To determine whether graphs are Hamiltonian and/or Eulerian.
4. To solve problems involving vertex and edge coloring.
5. To model real world problems using graph theory.

Course Contents:

UNIT I

Basics- Graphs, Degree sequences, Distance in graphs, Complete, Regular and bipartite graphs, Basic properties. Structure and Symmetry- Cut vertices, Bridges and blocks, Automorphism groups, Reconstruction problem.

UNIT II

Trees and connectivity - Properties of trees, Arboricity, Vertex and edge connectivity, Mengers theorem.

UNIT III

Eulerian and Hamiltonian graphs – Characterization of Eulerian graphs, Sufficient conditions for Hamiltonian graphs.

UNIT IV

Colouring and planar graphs - Vertex and edge colouring, Perfect graphs, Planar graphs, Euler's theorem, Kuratowski's theorem, Colouring of planar graphs, Crossing number and thickness.

UNIT V

External Graph theory - Turan's theorem, Ramsey's theorem, Szemeredi's regularity lemma, applications.

Text Books:

1. J. A. Bondy, U. S. R. Murthy, ***"Graph Theory"***, Springer; 1st Corrected ed. 2008, Corrected 3rd printing 2008 ed. (27 September 2011).
2. D. B. West, ***"Introduction to Graph Theory"***, Pearson Education India, 2nd edition,1st January 2015.

Reference Book:

1. Reinhard Diestel, "***Graph Theory"***, Springer Verlag, 5th Edition, 2017.

Course Title:	**Programming in Java**	**Semester**	**V**
Course Code	**BTITOE504C**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3– 0 – 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
2. To understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
3. To be able to use the Java SDK environment to create, debug and run simple Java programs.

Course Outcomes:

After learning the course, the students should be able:

1. To know the structure and model of the Java programming language.
2. To use the Java programming language for various programming technologies.
3. To develop software in the Java programming language (application).

Course Contents:

UNIT I

Introduction to Java: Fundamentals of Object-oriented Programming, Evolution of Java, Overview of Java Language: Data types in Java, Operators and expressions, Decision Making and Branching: Control Statements such as If Else, Do statement, For statement, The Else if ladder, Jumps in loops, Labelled loops, While repetition statement, Switch statement, Break and continue statement, Arrays, Strings and Vectors: Creating one dimensional and multidimensional array, Strings, Vectors, Wrapper classes, Enumerated types, Annotations.

UNIT II

Object Oriented Programming: Classes, Objects and Methods: Defining class, Methods, Creating objects, Accessing Class members, Static Methods, Finalize Methods, Visibility Control, Method overloading, Method Overriding, Recursion. Interfaces, Constructors and finalizes Methods.
Packages and Applet Programming: Java API Packages, Using System Packages, Naming conventions, Creating Packages and Jar Files, Accessing and using a package, Hiding Classes, Applet Programming.

UNIT III

Multithreading: Creating threads, Extending thread class, Stopping and Blocking a thread, Life cycle of a thread, Using thread method, Thread exceptions, Implementing the Run able interface, Inter thread communication. Managing Errors and Exceptions: Types of errors, Exceptions, Syntax of exception handling code, Multiple catch statements, Throwing your own exception, Using exceptions for debugging.

UNIT IV

Graphics Programming: The Graphics class, Lines and Rectangles, Circles, Arc and ellipses, Polygons, Drawing Bar charts, AWT Package and Swings.

UNIT V

Managing Files & I/O Handling: Files and Streams, Stream classes, Byte Stream Classes, Character Stream Classes, Using Streams, Reading / writing bytes and characters, Interactive Input and Output, Other Stream classes.

Text Books:

1. E. Balagurusamy, **"*Programming with Java – A Primer*"**, Tata McGraw-Hill Publication, 6th Edition, 2019.
2. Steven Holzner et al. **"*Java 2 Programming*"**, Black Book, Dreamtech Press, 2009.

Reference Books:

1. H.M. Deitel, P.J. Deitel, **"*Java - How to Program*"**, PHI Publication, 6th Edition, 2005.
2. Bruce Eckel, **"*Thinking in Java*"**, Pearson, 2008.
3. Tim Lindholm, Frank Yellin, Bill Joy, Kathi Walrath, **"*The Java Virtual Machine Specification*"**, Addison Wesley Publication, 8th Edition, 2014.

Course Title:	**Human Computer Interaction**	**Semester**	**V**
Course Code	**BTITOE504D**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3 – 0 – 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

Upon successful completion of this course, students should be able:

1. To design, implement and evaluate effective and usable graphical computer interfaces.
2. To describe and apply core theories, models and methodologies from the field of HCI.
3. To describe and discuss current research in the field of HCI.
4. To implement simple graphical user interfaces using the Java Swing toolkit.
5. To describe special considerations in designing user interfaces for older adults.

Course Outcomes:

After learning the course, the students should be able:

1. To describe and apply core theories, models and methodologies from the field of HCI.
2. To describe what the user-centred design cycle is and explain how to practice this approach to design interactive software systems.
3. To analyze the main features of interactive systems, and explain how to gauge the usability of digital environments, tools and interfaces.

Professional Skills:

1. Conduct user and task analysis.
2. Implement graphical user interfaces with modern software tools.
3. Critique and evaluate interactive software using guidelines from human factor theories.

Course Contents:

UNIT I

Introduction: Course objective and overview, Historical evolution of the field, The Human, The Computer, The Interaction.

UNIT II

Design processes: Interaction Design basics, Concept of usability – definition and elaboration, HCI in the Software Process, Design Rules.

UNIT III

Implementation and Evaluation: Implementation Support, Evaluation Techniques, Universal Design, Use Support.

UNIT IV

Models: Cognitive Models, Socio-Organizational Issues and Stakeholders Requirements, Communication and Collaboration models. Theories: Task Analysis Dialog notations and Design Models of the system Modelling Rich Interactions.

UNIT V

Modern Systems: Group ware, Ubiquitous Computing and Augmented Realities, Hypertext, Multimedia and World Wide Web.

Text Books:

1. Alan Dix, Janet Finlay, ***"Human Computer Interaction"***, Pearson Education, 3rd edition, 2009.
2. Ben Shneiderman, ***"Designing the User Interface - Strategies for Effective Human Computer Interaction"***, Pearson Education, 2010.

Reference Books:

1. M. B. Rosson, J. M. Carroll, ***"Usability Engineering: Scenario-Based Development of Human-Computer Interaction"***, Elsevier, 2002.
2. Alan Cooper, ***"The Essentials of Interaction Design"***, Wiley Publishing, 2007.
3. Nielsen, J. Morgan Kaufmann, San Francisco, **"*Usability Engineering"***, 1993.
4. Heim, S., **"*The Resonant Interface: HCI Foundations for Interaction Design"***, Addison-Wesley, 2007.

Course Title:	**Game Theory**	**Semester**	**V**
Course Code	**BTITOE504E**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3 – 0– 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. The course is intended for students and teachers of institutions which offer undergraduate engineering programmes.
2. The aim of the course is to provide an introduction to the study of game theory which has found wide applications in economics, political science, sociology, engineering apart from disciplines like mathematics and biology.
3. The course would introduce to the fundamental tools of game theory, a few equilibrium concepts, apart from numerous exercises and applications.
4. Knowledge of game theory would help students to understand and analyze real life situations such as market behavior or voting in elections, apart from equipping them with analytical concepts which might be useful for decision to pursue social sciences, engineering, sciences or managerial higher studies.
5. This is an interdisciplinary course, hence not only social sciences but science and engineering departments of different universities can benefit from it.

Course Outcomes:

After learning the course, the students should be able:

1. To solve problems using basic graph theory.
2. To identify induced sub graphs, cliques, matching's, covers in graphs.
3. To determine whether graphs are Hamiltonian and/or Eulerian.
4. To solve problems involving vertex and edge coloring.
5. To model real world problems using graph theory.

Course Contents:

UNIT I

Introduction to Game Theory: Concept of game theory, Theory of rational choice, Interacting decision makers. Strategic Games and Nash Equilibrium: Strategic games: examples, Nash equilibrium: concept and examples, Best response functions, Dominated Actions, Symmetric games and symmetric equilibrium.

UNIT II

Illustrations of Nash Equilibrium: Cournot's model of duopoly market, Bertrand's model of duopoly market, Electoral Competition, War of Attrition, Auctions, Accident Laws.

UNIT III

Mixed Strategy Nash EquilibriumL: Introduction, Strategic games with randomisation, Mixed strategy Nash equilibrium: concept and examples, Dominated Actions, Formation of Players' beliefs.

UNIT IV

Extensive Games and Nash Equilibrium: Introduction to extensive games, Strategies and outcomes, Nash equilibrium, Subgame perfect Nash equilibrium, Backward induction.

UNIT V

Illustrations of Extensive Games and Nash Equilibrium: Stackelberg model of duopoly markets, Ultimatum game.

Text books:

1. Osborne, M.J., "***An Introduction to Game Theory***", Oxford University Press, 2004
2. Mas-Colell, A., M.D. Whinston and J.R. Green, "***Microeconomic Theory***", Oxford University Press, 1995

Reference Book:

1. Gibbons, R., "***A Primer in Game Theory***", Pearson Education, 1992.

Course Title:	**3D Printing and Design**	**Semester**	**V**
Course Code	**BTITOE504F**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3 – 0– 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objective:

1. To impart knowledge and skills related to 3D printing technologies, selection of material and equipment and develop a product using this technique in Industry 4.0 environment.

Course Outcomes:

After learning the course, the students should be able:

1. To develop CAD models for 3D printing.
2. To import and Export CAD data and generate .stl file.
3. To select a specific material for the given application.
4. To select a 3D printing process for an application.
5. To produce a product using 3D Printing or Additive Manufacturing (AM).

Course Contents:

UNIT I

3D Printing (Additive Manufacturing): Introduction, Process, Classification, Advantages, Additive V/s Conventional Manufacturing processes, Applications.
CAD for Additive Manufacturing: CAD Data formats, Data translation, Data loss, STL format.

UNIT II

Additive Manufacturing Techniques: Stereo- Lithography, LOM, FDM, SLS, SLM, Binder Jet technology. Process, Process parameter, Process Selection for various applications.
Additive Manufacturing Application Domains: Aerospace, Electronics, Health Care, Defence, Automotive, Construction, Food Processing, Machine Tools.

UNIT III

Materials: Polymers, Metals, Non-Metals, Ceramics. Various forms of raw material- Liquid, Solid, Wire, Powder; Powder Preparation and their desired properties, Polymers and their properties, Support Materials.

UNIT IV

Additive Manufacturing Equipment: Process Equipment- Design and process parameters, Governing Bonding Mechanism, Common faults and troubleshooting, Process Design.

UNIT V

Post Processing: Requirement and Techniques. Product Quality: Inspection and testing, Defects and their causes.

Text books:

1. Lan Gibson, David W. Rosen and Brent Stucker, "***Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing***", Springer, 2010.
2. Andreas Gebhardt, "***Understanding Additive Manufacturing: Rapid Prototyping, RapidTooling, Rapid Manufacturing***", Hanser Publisher, 2011.
3. Dr. Sabrie Soloman, "***3D Printing and Design***", the 4th Industrial Revolution, 2020.
4. CK Chua, Kah Fai Leong, "***3D Printing and Rapid Prototyping- Principles and Applications***", World Scientific, 2017.
5. J.D. Majumdar and I. Manna, "***Laser-Assisted Fabrication of Materials***", Springer Series in Material Science, 2013.

Reference Books:

1. L. Lu, J. Fuh and Y.S. Wong, "***Laser-Induced Materials and Processes for Rapid Prototyping***", Kulwer Academic Press, 2001.
2. Zhiqiang Fan and Frank Liou, "***Numerical Modelling of the Additive Manufacturing (AM)Processes of Titanium Alloy***", InTech, 2012.

Course Title:	**Computer Networks and Internetworking Protocols Lab**	**Semester**	**V**
Course Code	**BTITL505**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 – 2**
Stream	**Core**	**Credit**	**1**

Lab Experiments List:

1. Conversion of IP addresses
   (e.g. I/P: 10.24.164.254 O/P: 00001010.00011000.10000000.11111110 and I/P:binary dotted O/P: decimal dotted)
2. Introduction to Wireshark
3. Wireshark Lab: Ethernet and ARP
4. Wireshark Lab: IP
5. Wireshark Lab: ICMP, study of ping and traceroute commands
6. Wireshark Lab: UDP
7. Wireshark Lab: TCP
8. Study of ftp, telnet tools and network configuration files
9. DHCP server configuration
10. Socket programming for UDP and TCP
11. Study of Network commands

Course Title:	**Software Engineering Lab**	**Semester**	**V**
Course Code	**BTITL506**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 –2**
Stream	**Core**	**Credit**	**1**

Lab Experiments List:

Part: I RDBMS

1. To develop a mini project for an RDBMS, the following exercise have been specified to give idea/ prerequisite learning for the concept required in defining the problem statement for an RDBMS.
2. Design and draw an ER/EER diagram and map this diagram to the database tables.
3. Create database tables for the problem. Perform add, insert, delete, update operations.
4. Use DDL statements and apply all constraints on tables to make the operations on tables.
5. Write and execute triggers and procedures/functions.
6. Generate a simple report.

Part: II

Following exercise has been specified to give idea/prerequisite learning for the concept in defining the problem statement for a front end to RDBMS based system.

Front end tools support developments of the following concept:

1. Controls
2. Properties for every control of the form
3. Events
4. Programming components
5. Proper interface to the back end database

The desktop database can be chosen from MSSQL, ORACLE, mySQL or equivalent databases packages. The front end development tools can be chosen from .NET, JAVA or equivalent tools.

Part: III

The statement of the problem will be the mini project for the group. The design of the project shall follow the software development life cycle. It should prepare a report for each stage (this will be the part of project manual later).

The group should understand and prepare proper documentation in relation with following:

1. Problem definition in detail.
2. Literature survey.
3. Requirement analysis.
4. System analysis (Draw Level 2 DFD at least).
5. System design
6. Implementation
7. Use cases
8. Testing

While designing the project the care should be taken to follow the coding conventions, software project design standards, data dictionary, etc. Staff in-charge will frame the mini project specification to be performed by group of students. There will be different problem definition to each group. The students will prepare an installable CD for the mini projects; *README* file will have the project description, system requirements, development details, and installation instruction. *User manual* will have the interaction screens and the way to use the developed project.

Course Title:	**Embedded Systems Lab**	**Semester**	**V**
Course Code	**BTITL506**	**Course Type**	**Elective**
Prerequisite	**Microprocessor and Microcontroller**	**L – T – P**	**0 – 0 – 2**
Stream	**Software Application & Development**	**Credit**	**1**

Lab Experiments List:

1. Study of ARM evaluation system
2. Interfacing ADC and DAC
3. Interfacing LED and PWM
4. Interfacing real time clock and serial port
5. Interfacing keyboard and LCD
6. Interfacing EPROM and interrupt
7. Mailbox
8. Interrupt performance characteristics of ARM and FPGA
9. Flashing of LEDS
10. Implementing zigbee protocol with ARM

Course Title:	**IT Service Management Lab**	**Semester**	**V**
Course Code	**BTITL506**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 – 2**
Stream	**Infrastructure & Security Management**	**Credit**	**1**

Lab Experiments List:

1. To study the Information System Design Process.
2. To study the relationship of service level management with other service delivery processes.
3. To study the Problem, Change and Incident Management.
4. To study and demonstrate disaster recovery.
5. To study and demonstrate the various security techniques used to secure the data while transmitting over the internet.

Course Title:	**Information Storage Management Lab**	**Semester**	**V**
Course Code	**BTITL506**	**Course Type**	**Elective**
Prerequisite	**Computer Architecture & Organization**	**L - T - P**	**0- 0 - 2**
Stream	**Information Management & Quality Control**	**Credit**	**1**

Lab Experiments List:

1. Data Center Environment
   a) Install the VNXe Simulator
   b) Discover the infrastructure
2. Intelligent storage system
   a) Navigate the storage system
   b) Create a block device
   c) Create a file device
3. FC SAN
   a) FC san configuration
   b) FC san trace
4. IP SAN
   a) IP SAN configuration
   b) ISCSI san trace
5. Host-based business continuity
   a) Multipath
6. Managing protection services
   a) Array-based protection
   b) Configuring LUN protection
7. Managing storage infrastructure
   a) Monitoring and reporting

Course Title:	**Network Management Lab**	**Semester**	**V**
Course Code	**BTITL506**	**Course Type**	**Elective**
Prerequisite	**Computer Architecture & Organization**	**L – T – P**	**0– 0 – 2**
Stream	**Network**	**Credit**	**1**

Lab Experiments List:

1. Network Monitoring tools
   a) Status b) Route c) Traffic Tools
2. Monitoring and management network using SNMP
   a) Basic SNMP b) Advanced SNMP v3 Authentication/Encryption and ACL
   c) SNMP Trap Daemon Implementation
3. Install and configure SNMP MIB browser
   a) qtmib b) snmpB c) OpManager MIB browser
4. Network Statistics and measurement
   a) LAN Traffic Monitoring b) Protocol statistics
5. LAN Troubleshooting using Wireshark.
6. To study log system using open source tools.
7. Study of commercial network management tools: HPOpenView, Orphanage, GFILanguard and IBM NMS.

Course Title:	**Data Visualization Lab**	**Semester**	**V**
Course Code	**BTITL506**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 – 2**
Stream	**Data Science**	**Credit**	**1**

Instructions: It is proposed to use the various datasets from standard data repositories such as Kaggle, UCI, Git hub and stock market websites such as BSE. Minimum ten programs/notebooks are required to be completed by the students. Following are the model experiments. However, the course-in-charge can modify these experiments in order to cover all the topics of Data Visualization course.

Lab Experiments List:

1. Program/Notebook to demonstrate area plots and how to create them with Matplotlib, histograms with Matplotlib
2. Pandas Data Manipulation
   A) Pandas
   B) Series
   C) Timedelta
3. Program/Notebook to draw scatter plots, bubble plots, racing bar chart with Matplotlib.
4. Program/Notebook related to any dataset using Pandas DataFrame methods:
   Count ( ) Method, describe ( ) Method, drop_duplicated( ) Method, empty property, filter( ) Method, equals( ) Method
5. To learn about advanced visualization tools such as waffle charts and word clouds and how to create them.
6. To learn about seaborn, visualization library, and how to use it to generate attractive regression plots.
7. To learn about Folium, visualization library, designed especially for visualizing geospatial data.
8. To learn how to use Folium to create maps of different regions of the world and how to superimpose markers on top of a map, and how to create choropleth maps.
9. Minimum three case studies (such as Loan prediction, Counties population data analysis, COVID-19 and Stock market data analysis etc)
10. Program/Notebook case study using Seaborn and Scikit learn library

.

Course Title:	**Virtual Reality Lab**	**Semester**	**V**
Course Code	**BTITL506**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 – 2**
Stream	**---**	**Credit**	**1**

Lab Experiments List:

1. Developing architecture of a house using Virtual Reality
2. Perform CRO based experiment using Virtual Reality
3. Undertaking qualitative analysis in Chemistry using Virtual Reality
4. Carry out assembly/disassembly of an engine using Virtual Reality
5. Explore human anatomy using Virtual Reality
6. Simulation of circulation of blood in heart
7. Simulation of Fight/Vehicle/Space Station
8. Building Electronic circuit using Virtual Reality, given basic electronic components
9. Developing concept of Virtual class room with multiplayer

Course Title:	**Operating Systems**	**Semester**	**VI**
Course Code	**BTITC601**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**3 – 1 – 0**
Stream	**Core**	**Credits**	**4**

Course Objectives:

1. To study the basic concepts and functions of operating systems.
2. To understand the structure of operating systems.
3. To learn about Processes, Threads and Scheduling algorithms.
4. To understand the principles of Concurrency and Deadlocks.
5. To learn various memory management schemes.
6. To study File systems.

Course Outcomes:

After learning the course, the students should be able:

1. To design various Scheduling algorithms.
2. To apply the principles of concurrency.
3. To design deadlock, prevention and avoidance algorithms.
4. To compare and contrast various memory management schemes.
5. To design and Implement a prototype file system.

Course Contents:

UNIT I

Operating System Structures: Definition, Types of operating system, System components, System services, Systems calls, System programs, System structure, Virtual machines, System design and implementation.

UNIT II

Processes and CPU scheduling: Process concept, Process scheduling, Operation on a process, Co-operating processes, Threads, Interprocess communication, Scheduling criteria, Scheduling algorithms, Multiple-processor scheduling, Real-time scheduling, Scheduling algorithms and performance evaluation.

Process Synchronization: The critical-section problem, Critical regions, Synchronization hardware, Semaphores, Classical problems of synchronization, Monitors.

UNIT III

Deadlocks: Systems model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock, combined approach to deadlock handling.

UNIT IV

Memory Management and Virtual Memory: Logical versus physical address space, Swapping, Contiguous allocation, Paging, Segmentation with paging, Demand paging, Page replacement algorithms, Thrashing.

UNIT V

File Management: File system and secondary storage devices, Real-time operating systems.

Text Books:

1. A. Silberschatz, P. Galvin, ***"Operating System Concepts"***, Wiley Publication, 10th Edition, 2018.
2. A. S. Tanenbaum, H. Bos, ***"Modern Operating Systems"***, Pearson Education, 4th Edition, 2016.

Reference Books:

1. D.M. Dhamdhere,***"Systems Programming and Operating Systems"***, Tata McGraw Hill Publication, 2nd Edition, 2001.
2. G. Nutt, ***"Operating Systems Concepts"***, Addison Wesley Publication, 3rd Edition.
3. H. M. Deitel, ***"An Introduction to Operating Systems"***, Pearson education Publication, 3rd Edition, 2007.

Course Title:	**Database Management Systems**	**Semester**	**VI**
Course Code	**BTITC602**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**3 – 1– 0**
Stream	**Core**	**Credits**	**4**

Course Objectives:

1. To understand architecture and functioning of database management systems.
2. To learn relational model.
3. To use structured query language (SQL) and its syntax, transactions, database recovery and techniques for query optimization.
4. To acquaint with various normalization forms and query processing.
5. To learn indexing methods.

Course Outcomes:

After learning the course, the students should be able:

1. To explain need of database management.
2. To design and implement a database schema for a given problem-domain.
3. To normalize a database.
4. To create and query a database using SQL DML/DDL commands, stored procedures and functions.
5. To declare and enforce integrity constraints on a database.
6. To illustrate understanding of indexing methods.

Course Contents:

UNIT I

Introduction: Basic concepts, Advantages of DBMS over file-processing systems, Data abstraction, Data models and data independence, Components of DBMS and overall structure of DBMS, Data modeling, Entity, Attributes, Relationships, Constraints, Keys E-R diagrams, Components of E-R Model.

UNIT II

Relational Model: Basic concepts, Attributes and domains, Concept of integrity and referential constraints, Schema diagram, Relational query languages, Relational Algebra and Relational Calculus: Tuple relational and domain relational calculus.

Structured Query Language-I: Introduction, Characteristics and advantages, Data types and literals, DDL, Tables: creating, modifying, deleting.

UNIT III

Views: Creating, Dropping, Updation using views, DML, Operators, SQL DML queries, SELECT query and clauses.

Structured Query Language- II: Set operations, Predicates and joins, Set membership, Tuple variables, Set comparison, Ordering of tuples, Aggregate functions, Nested queries, Database modification using SQL Insert, Update and Delete queries, Dynamic and SQL and concept of stored procedures, Query-by-example.

UNIT IV

Relational Database Design: Notion of normalized relations, Functional dependency, Decomposition and properties of decomposition, Normalization using functional dependency, Multi-valued dependency and join dependency. Storage and File Systems: Secondary storage, RAID, File organization, Indices, Static and dynamic hashing, B-Trees and B+ Trees.

UNIT V

Query Processing and Transaction Management: Measures of query cost, Selection operation, Sorting and join operation, Transaction concept, Components of transaction management, Concurrency and recovery system, Different concurrency control protocols such as timestamps and locking, Validation, Multiple granularity, Deadlock handling, Different crash recovery methods such as log-based recovery, Shadow-paging, Buffer management and Remote backup system.

Text Books:

1. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, *"Database System Concepts"*, McGraw Hill Education, 6th Edition, 2011.

2. RamezElmasri and Shamkant B. Navathe, *"Fundamental Database Systems"*,Pearson Education, 7th Edition, 2015.

3. Raghu Ramkrishnan, Johannes Gehrke, *"Database Management Systems"*, McGraw Hill Education, 3rd Edition, 2007.

Reference Books:

1. Carlos Coronel, Steven Morris ***"Database systems: Design Implementation andManagement"***, Cengage Learning Press, 11th Edition, 2014.
2. J. Murach, ***"Murach's MySQL"***, Shroff Publication, 2nd Edition, 2016.
3. J. Murach, ***"Murach's Oracle SQL and PL/SQL: Works with All Versions Through 11g"***, Shroff Publication, 2008.

Course Title:	**Software Testing**	**Semester**	**VI**
Course Code	**BTITPE603A**	**Course Type**	**Elective**
Prerequisite	**Software Engineering**	**L – T – P**	**3 – 0 – 0**
Stream	**Software Application & Development**	**Credits**	**3**

Course Objectives:

1. To study fundamental concepts in software testing, including software testing objectives, processes, criteria, strategies, and methods.
2. To learn planning of a test project, designing test cases and test data, conducting test operations, managing software problems and defects, and generating a test report.
3. To develop an understanding of the meaning and importance of quality in relation to software systems and the software development process.
4. To study issues and techniques for implementing and managing software quality assurance processes and procedures.

Course Outcomes:

After learning the course, the students should be able:

1. To apply software testing knowledge and its processes to software applications.
2. To identify various software testing problems.
3. To solve software testing problems by designing and selecting software test models, criteria, strategies and methods.
4. To apply the techniques learned to improve the quality of software development.
5. To prepare a software quality plan for a software project.

Course Contents:

UNIT I

Principles of Testing Software development life cycle model: Phases of software project, Quality, Quality assurance and quality control, Testing, Verification and validation, Process models to represent various phases, Life cycle models, Software testing life cycle.

White Box Testing (WBT) and Black Box Testing: Static testing, Structural testing, Challenges in WBT. Black box testing: Black box testing process.

UNIT II

Integration Testing: Definition, As a type of testing: Top-down integration, Bottom-up integration, Bi-directional integration, System integration, Choosing integration method, As a phase of testing, Scenario testing: System scenarios, Use case scenarios, Defect bash.

UNIT III

System and Acceptance Testing, Functional Vs non Functional, Functional system testing, Non-functional system testing, Acceptance testing.

UNIT IV

Performance testing, Regression testing, Internationalization testing, Adhoc testing. Factors governing performance of testing, Methodology, tools and process for performance testing. Regression Testing: Introduction, Types of Regression testing, Regression testing process. Adhoc testing: Introduction, Buddy testing, Pair testing, Exploratory testing, Iterative testing, Agile and Extreme testing, XP work flow, Defect seeding.

UNIT V

Testing Object Oriented Software: Introduction, Comparison of object oriented and procedural software, System testing example, Unit testing of classes, Tools for testing object oriented software, Testing web applications.

Text Book:

1. Srinivasan Desikan, Gopalaswamy Ramesh, ***"Software Testing: Principles and Practices"***, Pearson publication, 2nd Edition, 2006.

Reference Books:

1. Louise Tamres, ***"Introducing Software Testing"***, Pearson publication, 2002.
2. Boris Beizer, ***"Software Testing Techniques"***, Dreamtech press, 2nd Edition, 2014.

Course Title:	**Data Storage Technologies & Networks**	**Semester**	**VI**
Course Code	**BTITPE603B**	**Course Type**	**Elective**
Prerequisite	**Computer Network & Internetworking Protocols, Operating Systems**	**L – T – P**	**3 – 0 – 0**
Stream	**Infrastructure & Security Management**	**Credits**	**3**

Course Objectives:

1. To gain knowledge and understand the design of a Data Centre.
2. To learn the options in the running of an efficient Data Centre.
3. To understand the value of data to a business, information lifecycle.
4. To understand the challenges in data storage and data management.
5. To learn solutions available for data storage.

Course Outcomes:

After learning the course, the students should be able:

1. To explain the design of a data center and storage requirements.
2. To discuss the various types of storage and their properties.
3. To explain physical and virtualization of storage.
4. To explain the backup, archiving with regard to recovery and business continuity.

Course Contents:

UNIT I

Data Centre: Introduction, Site Selection and Environmental Considerations, Hierarchical or Layered Architecture, Architect Roles, Goals and Skills, Architecture Precursors.

Data Centre Design: Architecture Design and Standards Recommendations, Raised Access Floor and Design Best Practices, Connecting the infrastructure with copper and fiber. IT Hardware, Cooling System Options and Environmental Control, Electrical Power Systems, Room Layout, Fire Protection and Security Systems, Building Automation and Energy Management Systems, Commissioning and Handover.

UNIT II

Storage Management: Introduction to Storage Technology, Storage Systems Architecture, Physical and logical components of a connectivity environment, Major physical components of a disk drive and their functions, Concept of RAID and its components, Different RAID levels and their suitability for different application environments: RAID 0, RAID 1, RAID 3, RAID 4, RAID 5, RAID 0+1, RAID 1+0, RAID 6, Integrated and Modular storage systems, high-level architecture and working of an intelligent storage systems.

UNIT III

Networked Storage: Evolution of networked storage, Architecture, components, and topologies of FC-SAN, NAS, and IP-SAN, Benefits of the different networked storage options, Need for long-term archiving solutions and describe how CAS fulfill the need, Appropriateness of the different networked storage options for different application environments.

UNIT IV

Managing Data Center: Reasons for planned/unplanned outages, Impact of downtime, Difference between business continuity (BC) and disaster recovery (DR), RTO and RPO, Identification of single points of failure in a storage infrastructure and solutions to mitigate these failures, Architecture of backup/recovery and the different backup/recovery topologies, replication technologies and their role in ensuring information availability and business continuity Remote replication technologies and their role in providing disaster recovery and business continuity capabilities, Key areas to monitor in a data center, Industry standards for data center monitoring and Management Key metrics to monitor storage infrastructure.

UNIT V

Securing Storage and Storage Virtualization: Information Security, Critical security attributes for information systems, Storage security domains, Analyze the common threats in, each domain, Storage Virtualization: Forms, Configurations and Challenges, Types of Storage Virtualization: Block-level and File-Level.

Text Books:

1. Mauricio Arregoces, ***"Data Center Fundamentals"***, Cisco Press, 1st edition, 2003.
2. Robert Spalding, ***"Storage Networks: The Complete Reference"***, Tata McGraw Hill, Osborne, 2003.
3. Marc Farley, ***"Building Storage Networks"***, Tata McGraw Hill, Osborne. 2001.
4. Meeta Gupta, ***"Storage Area Network Fundamentals"***, Pearson Education Limited, 2002.

Reference Books:

1. G. Somasundaram, Alok Shrivastava, ***"Information Storage and Management"***, EMC Education Series, Wiley Publishing Inc., 2011.
2. Gustavo Santana, ***"Data Center Virtualization Fundamentals: Understanding Techniques and Designs for Highly Efficient Data Centers with Cisco Nexus, UCS, MDS, and Beyond"***, Cisco Press, 1st Edition, 2013.

Course Title:	**Service Oriented Architecture**	**Semester**	**VI**
Course Code	**BTITPE603C**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L - T - P**	**3 - 0 - 0**
Stream	**Information Management & Quality Control**	**Credits**	**3**

Course Objectives:

1. To gain understanding of the basic principles of service orientation.
2. To learn service oriented analysis techniques.
3. To learn technology underlying the service design.
4. To learn advanced concepts such as service composition, orchestration and choreography.
5. To know about various WS specification standards.

Course Outcomes:

After learning the course, the students should be able:

1. To build applications based on XML.
2. To develop web services using technology elements.
3. To build SOA-based applications for intra-enterprise and inter-enterprise applications.

Course Contents:

UNIT I

Introducing SOA: Fundamental SOA: Common Misperceptions about SOA, Common tangible benefits of SOA, Common pitfalls of adopting SOA, The Evolution of SOA: -from XML to Web services to SOA, The continuing evolution of SOA, The roots of SOA. Web Services and Primitive SOA: The Web services framework-Services, Service descriptions, Messaging with SOAP.

UNIT II

Web Services and Contemporary SOA: Message exchange patterns, Service activity, coordination, Atomic transactions, Business activities, Orchestration, Choreography- Web Services and Contemporary SOA: Addressing, Reliable messaging, Correlation.

Policies Metadata exchange: Security, Notification and eventing and Service-Orientation: principles of Service - Anatomy of a service-oriented architecture, Common principle of service orientation, Service Layers, Service orientation.

UNIT III

Building SOA: SOA Delivery Strategies, SOA delivery lifecycle phases. Service-Oriented Analysis: Introduction to service-oriented analysis-Benefits of a business centric SOA- Deriving business service, Service modeling, Service modeling guidelines, Classifying service model logic, Contrasting service modeling approaches.

UNIT IV

Service-Oriented Design: Introduction to service-oriented design, WSDL-related XML Schema language basics, WSDL language basics, SOAP language basics, Service interface, Design tools. SOA

Composition Guidelines: Steps to composing SOA, Considerations for choosing service layers and SOA standards, positioning of cores and SOA extensions.

UNIT V

SOA Service Design: - Overview-Service design of business service, Application service, Task centric service and guidelines, SOA Business Process Design: WS-BPEL language basics, WS Coordination,

SOA support in J2EE - Java API for XML-based web services (JAX-WS) , Java architecture for XML binding (JAXB) , Java API for XML Registries (JAXR) , Java API for XML based RPC (JAX-RPC), Web Services Interoperability Technologies (WSIT).

Text Books:

1. Thomas Erl, "***Service-Oriented Architecture: Concepts, Technology, and Design"***, Pearson Education, 2016.
2. Frank. P. Coyle, "***XML, Web Services and The Data Revolution"***, Pearson Education, 2002.
3. Sandeep Chatterjee, James Webber, "***Developing Enterprise Web Services. An Architect's Guide"***, Pearson Education, 2005.
4. Eric Newcomer, Greg Lomow, "***Understanding SOA with Web Services"***, Pearson Education, 2005.
5. Schmelzer et al., "***XML and Web Services"***, Pearson Education, 2002.

Reference Books:

1. Dan woods and Thomas Mattern, ***"Enterprise SOA: designing IT for Business Innovation"***, O'REILLY, 2008.
2. James McGovern, Sameer Tyagi, Michael E. Stevens, Sunil Mathew, "***Java Web Services Architecture"***, Morgan Kaufmann Publishers, 2003.
3. AtulKahate, "***XML and Related technologies"***, Pearson Education, 2008.
4. Kennard Scibner and Mark C. Stiver, "***Understanding SOAP"***, SAMS publishing.
5. B. V. Kumar, S. V. Subrahmanya, "***Web Services: An Introduction"***, TMH India, 2nd Edition, 2012.

Course Title:	**Network Programming**	**Semester**	**VI**
Course Code	**BTITPE603D**	**Course Type**	**Elective**
Prerequisite	**Computer Network & Internetworking Protocols, Operating Systems**	**L – T – P**	**3 – 0 – 0**
Stream	**Network**	**Credits**	**3**

Course Objectives:

1. To learn the basics of socket programming using TCP Sockets.
2. To learn about Socket options.
3. To learn to develop Macros for including objects in MIB Structure.
4. To understand SNMP v1, v2 and v3 protocols & practical issues.

Course Outcomes:

After learning the course, the students should be able:

1. To analyze the requirements of a networked programming environment and identify the issues to be solved.
2. To create conceptual solutions to those issues and implement a programming solution.
3. To understand the key protocols that supports the Internet.
4. To apply several common programming interfaces to network communication.
5. To understand the use of TCP/UDP Sockets.
6. To apply advanced programming techniques such as Broadcasting, Multicasting.

Course Contents:

UNIT I

Socket and Application Development: Introduction to Socket Programming, System Calls, Address conversion functions, POSIX Signal Handling, Server with multiple clients, Boundary conditions, Server process Crashes, Server host Crashes, Server Crashes and reboots, Server Shutdown, I/O Multiplexing, I/O Models, TCP echo client/server with I/O Multiplexing.

UNIT II

Socket Option: Socket options, getsockopt and setsockopt functions, Generic socket options, IP socket options, ICMP socket options, TCP socket options, Multiplexing TCP and UDP sockets, SCTP Sockets, SCTP Client/server, Streaming Example, Domain name system, gethostbyname, gethostbyaddr, getservbyname and getservbyport functions, Protocol Independent functions in TCP Client/Server Scenario.

UNIT III

Advanced Socket: IPv4 and IPv6 interoperability, Threaded servers, Thread creation and termination, TCP echo server using threads, Mutex Condition variables, Raw sockets, Raw socket creation, Raw socket output, Raw socket input, ping program, traceroute program.

UNIT IV

Simple Network Management: SNMP network management concepts, SNMPv1 Management information, MIB Structure, Object syntax, Standard MIB's, MIB-II Groups, SNMPv1 protocol and Practical issues.

SNMPv2, SNMPv3 and RMON: Introduction to SNMPv2, SMI for SNMPv2 Protocol, SNMPv3 Architecture and applications, Security and access control model, Overview of RMON.

UNIT V

Protocols, Sessions, State, and Implementing Custom Protocols State vs. Stateless, Methods for Maintaining State, What is a Protocol? Designing a Custom Protocol, Our Chat Protocol, Protocol Registration, Elementary Name, Address Conversions and design decisions, Domain Name System, gethostbyname function, RES_USE_INET6 Resolver Option, gethostbyname2 function and IPv6 Support, gethostbyaddr function, uname function, gethostname function, getservbyname and getservbyport functions

Text Books:

1. W. Richard Stevens, ***"UNIX Network Programming Vol-I"***, Addison-Wesley Professional, 3rd Edition,2003.
2. William Stallings, ***"SNMP, SNMPv2, SNMPv3 and RMON 1 and 2"***, Pearson Edition, 3rd Edition, 2009.

Reference Book:

1. D.E. Comer, ***"Internetworking with TCP/IP Vol- III: Client-Server Programming and Application BSD Sockets Version"***, Pearson Edition, 2nd Edition, 2003.

Course Title:	**Data Warehousing and Data Mining**	**Semester**	**VI**
Course Code	**BTITPE603E**	**Course Type**	**Elective**
Prerequisite	**Database Management Systems**	**L – T – P**	**3 – 0 – 0**
Stream	**Data Science**	**Credits**	**3**

Course Objectives:

1. To introduce the concepts, techniques, design and applications of Data Warehousing and Data Mining.
2. To enable students to understand and implement classical algorithms in Data Warehousing and Data Mining.
3. To enable students to learn how to analyze the data, identify the problems and choose the relevant algorithms to apply.

Course Outcomes:

After learning the course, the student will be able:

1. To understand the functionality of the various Data Warehousing and Data Mining components.
2. To recognize the strengths and limitations of various Data Warehousing and Data Mining models.
3. To compare the various approaches to Data Warehousing and Data Mining implementations.
4. To describe and utilize a range of techniques for designing Data Warehousing and Data Mining systems for real-world applications.

Course Contents:

UNIT I

Introduction to data warehousing, Evolution of decision support systems, Modeling a data warehouse, Granularity in the data warehouse, Data warehouse life cycle, Building a data warehouse, Data Warehousing Components, Data Warehousing Architecture.

UNIT II

Online Analytical Processing, Categorization of OLAP Tools, Introduction to Data mining and knowledge discovery, Relation to Statistics, Databases, Data Mining Functionalities, Steps in Data Mining Process, Architecture of a Typical Data Mining Systems, Classification of Data Mining Systems.

UNIT III

Overview of Data Mining Techniques, Data Preprocessing, Data Cleaning, Data Integration, Data Transformation and Data Reduction, Data Generalization and Summarization Based Characterization, Mining Association Rules in Large Databases.

UNIT IV

Classification and Prediction, Issues regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Other Classification Methods.

Prediction, Clusters Analysis, Types of Data in Cluster Analysis, Categorization of Major Clustering Methods, Partitioning methods, Hierarchical methods.

UNIT V

Applications of Data Mining, Social Impacts of Data Mining, Case Studies, Mining WWW, Mining Text Databases, Mining Spatial Databases.

Text Books:

1. Adriaans, ***"Data mining"***, Addison- Wesley, 2009.
2. Margaret Dunham, ***"Data Mining: Introductory and Advanced Topics"***, Published by Prentice Hall.
3. Weiss, Sholom M., "***Predictive data mining: a practical guide",*** Kaufmann Publishers, 2008.

Reference Books:

1. Pang-Ning Tan, Michael Steinback, Vipin Kumar, ***"Introduction to Data Mining"***, Pearson Education, 2021.

2. M. Humphires, M. Hawkins, ***"Data Warehousing: Architecture and Implementation"***, Pearson Education, 2009.

3. Anahory, Murray, ***"Data Warehousing in the Real World"***, Pearson Education, 2008.

4. Kargupta, Joshi, et al., ***"Data Mining: Next Generation Challenges and Future Directions"***, Prentice Hall of India Pvt. Ltd, 2007.

Course Title:	**Compiler Design**	**Semester**	**VI**
Course Code	**BTITOE604A**	**Course Type**	**Elective**
Prerequisite	**Data Structures & Applications**	**L – T – P**	**3 – 0– 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To introduce the major concept areas of language translation and compiler design.
2. To develop an awareness of the function and complexity of modern compilers.
3. To provide practical, hands on experience in compiler design.

Course Outcomes:

After learning the course, the students should be able:

1. To understand the major concept areas of language translation and compiler design.
2. To develop an awareness of the function and complexity of compilers.
3. To identify the similarities and differences among various parsing techniques and grammar transformation techniques.

Course Contents:

UNIT I

Introduction to Compiling and Lexical Analysis: Definition, analysis of the source program, the phases of a compiler, the grouping of phases, Compiler-Construction tools, Role of the Lexical analyzer, Input buffering, Specification of Tokens, A Language for Specifying Lexical Analyzers, Design of a Lexical Analyzer generator.

UNIT II

Syntax Analysis: The role of the Parser, Context-free grammars, Writing a Grammar, Top-Down Parsing, Bottom- Up Parsing, Operator-precedence Parsing, LR-Parsers, Using Ambiguous Grammars, Parser Generators.

UNIT III

Syntax-Directed Translation: Definitions, Construction of Syntax Trees, Bottom-Up Evaluation of S Attributed definitions, Top-Down Translation, Bottom-Up Evaluation of Inherited attributes.

UNIT IV

Source language issues, parameter passing, symbol tables.
Intermediate Code Generation: Variants of syntax trees, Three address code, Types and declarations, Type checking, Control Flow and Backpatching, procedure calls.

UNIT V

Code Generation and Code Optimization: Code Generation: Issues in the design of code generation, The target language, Code Optimization: Need of code optimization, Principal sources of optimization, Basic blocks and flow graphs, Optimization of basic blocks, Peephole optimization.

Text Books:

1. Aho, Sethi, Ullman, ***"Compilers-Tools and Techniques"***, Pearson, 2nd Edition, 2015.
2. Tremblay, Sorenson, ***"Theory and Practice of Compiler Writing"***, McGraw Hill Publication.
3. Hopcroft, ***"Introduction to Automata Theory, Languages and Computation"***, Pearson Publication.

Reference Books:

1. Paul G. Sorenson, ***"Compiler Writing"***, Tata McGraw Hill.
2. Robin Hunter, ***"The Essence of Compilers"***, Pearson Publication, 2005.

Course Title:	**Enterprise Resource Planning**	**Semester**	**VI**
Course Code	**BTITOE604B**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**3 – 0– 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To introduce to enterprise systems and show how organizations use enterprise systems to run their operations more efficiently and effectively.
2. To learn about the critical success factors and implementation strategies that lead to enterprise system success.
3. To learn about the informational, knowledge, and decision-making opportunities afforded by enterprise systems.
4. To examine typical Enterprise Systems modules: Materials Management (MM), Supply Chain Management (SCM), Customer Relationship Management (CRM), Human Resource Management (HRM).

Course Outcomes:

After learning the course, the students should be able:

1. To demonstrate a good understanding of basic issues in Enterprise Systems.
2. To explain the scope of common Enterprise Systems (e.g., MM, SCM, CRM, HRM, procurement).
3. To explain the challenges associated with implementing enterprise systems and their impacts on organizations.
4. To describe the selection, acquisition and implementation of enterprise systems.
5. To use one of the popular ERP packages to support business operations and decision-making.
6. To communicate and assess an organization's readiness for enterprise system implementation with a professional approach in written form.
7. To demonstrate an ability to work independently and in a group.

Course Contents:

UNIT I

Enterprise Resource Planning: Introduction, Disadvantages of non-ERP systems, What Is ERP? Need of ERP, Advantage of ERP, Risks of ERP, Growth of ERP.

ERP Modules: Finance, Production Planning, Control and Management, Sales and Distribution, Human Resource Management, Inventory Control System, Quality Management, Plant Maintenance.

UNIT II

ERP Implementation: ERP Implementation (Transition) strategies, ERP Implementation Life Cycle, Implementation Methodologies, Evaluation and selection of ERP package, ERP Project Team: Vendors, Employees, Consultants, Training & Education, Project management & Monitoring, Post Implementation Activities, Operation & maintenance of ERP system, Measuring the Performance of ERP System, Success & failure factors of an ERP, Implementation.

UNIT III

ERP Market and Vendors: ERP Marketplace and Marketplace Dynamics, Comparison of Current ERP Packages and Vendors, like; SAP, Oracle, PeopleSoft, BAAN etc.

UNIT IV

ERP and Related Technologies: Business Process Re-Engineering (BPR), Information Systems - Management Information System (MIS), Decision Support System (DSS), Executive Support System (ESS), Data Warehousing, Data Mining, Online Analytical Processing (OLAP), Supply Chain Management, Customer Relationship Management.

UNIT V

ERP Case Studies: ERP systems implemented in: TISCO, SKF Automotive Bearings Co. Ltd, Qualcomm CDMA, California, Post Implementation review of ERP packages in Manufacturing, Services and Others Organizations, Customization of ERP for different types of Industries.

Text Books:

1. Alexis Leon, ***"ERP Demystified"***, TMH New Delhi, 3rd Edition.
2. V. K. Garg & N. K. Venkita Krishnan, ***"ERP Ware: ERP Implementation Framework"***, PHI.

Reference Book:

1. V. K. Garg & N. K. Venkita Krishna, ***"ERP Concepts & Planning"***, PHI, 2nd Edition.

Course Title:	**Decision Support Systems**	**Semester**	**VI**
Course Code	**BTITOE604C**	**Course Type**	**Elective**
Prerequisite	**Database Management Systems**	**L – T – P**	**3 – 0 – 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To select appropriate modeling techniques for supporting semi-structured business decision making.
2. To identify and select appropriate decision support systems for generating innovative business solutions.
3. To design and implement decision support systems for generating innovative business solutions.

Course Outcomes:

After learning the course, the students should be able:

1. To recognize the relationship between business information needs and decision making.
2. To know the general nature and range of decision support systems.
3. To understand issues related to the development of DSS.
4. To select appropriate modeling techniques.
5. To analyze, design and implement a DSS.

Course Contents:

UNIT I

Basic Concepts: Decision making systems, Modeling and support, Basics and definitions, Systems models, Modeling process, Decision making, Intelligence phase, Design phase, Choice phase, Evaluation, Implementation phase, Alternative decision making models, Decision support systems, Decision makers, Case applications.

UNIT II

Decision Support System Development: Decision support system development, Basics, Life cycle, Methodologies, Prototype, Technology levels and tools, Development platforms, Tool selection, Developing DSS, Enterprise systems, Concepts and definition, Evolution of information systems, Information needs, Characteristics and capabilities, Comparing and integrating EIS and DSS, EIS data access, Data warehouse, OLAP, Multidimensional analysis, Presentation and the Web, Including soft information enterprise on systems, Organizational DSS, Supply and value chains, Decision support, Supply chain problems and solutions, Computerized systems. MRP, ERP, SCM, Frontline decision support systems.

UNIT III

Knowledge Management: Organizational learning and memory, Knowledge management, Development methods, Technologies and tools, Success , Knowledge management and artificial intelligence, Electronic Document Management, Knowledge Acquisition and Validation, Knowledge

Engineering – Scope, Acquisition Methods, Interviews, Tracking Methods, Observation and other Methods, Grid Analysis, Machine Learning, Rule Induction, Case-Based Reasoning, Neural Computing, Intelligent Agents, Selection of an appropriate Knowledge Acquisition Methods, Multiple Experts, Validation and Verification of the Knowledge Base-Analysis, Coding, Documenting, and Diagramming.

UNIT IV

Knowledge Acquisition, Knowledge Acquisition and the Internet/Intranets, Knowledge Representation Basics, Representation in Logic and other Schemas, Semantic Networks, Production Rules, Frames, Multiple Knowledge Representation, Experimental Knowledge Representations, Representing Uncertainty. Intelligent System Development: Inference Techniques, Reasoning in Artificial Intelligence, Inference with Rules, Inference Tree, Inference with Frames, Model Based and Case Based Reasoning, Explanation and Meta Knowledge, Inference with Uncertainty, Representing Uncertainty, Probabilities and Related Approaches, Theory of Certainty, Approximate Reasoning using Fuzzy Logic,Intelligent Systems Development, Prototyping, Project Initialization, System Analysis and Design, Software Classification.

UNIT V

Building Expert Systems with Tools, Shells and Environments, Software Selection, Hardware, Rapid Prototyping and a Demonstration Prototype, System Development, Implementation, Post Implementation, Management Support Systems: Implementing and Integrating Management Support Systems, Implementation, Major Issues, Strategies, System Integration, Generic Models MSS, DSS–ES, Integrating EIS, DSS and ES, Global Integration, Intelligent DSS, Intelligent Modeling and Model Management, Examples of Integrated Systems, Problems and Issues in Integration.

Text Book:

1. Efrain Turban and Jay E. Aronson, "***Decision Support Systems and Intelligent Systems"***, Pearson Education, 7th Edition, 2005.

Reference Books:

1. Ganesh Natarajan and SandhyaShekhar, ***"Knowledge Management Enabling Business Growth"***, Tata McGraw Hill, 2002.

2. George M. Marakas, ***"Decision Support System"***, Prentice Hall, India, 2003.

3. Efrem A. Mallach, ***"Decision Support and Data Warehouse Systems"***, Tata McGraw Hill, 2002.

4. Kimiz Dalkir, ***"Knowledge Management: Theory and Practice"***, Elsevier Science, 2005.

5. Becerra Fernandez and Laidener, ***"Knowledge Management: An Evolutionary View"***, PHI, 2009.

Course Title:	**Software Project Management**	**Semester**	**VI**
Course Code	**BTITOE604D**	**Course Type**	**Elective**
Prerequisite	**Software Engineering**	**L – T – P**	**3 – 0 – 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To gain knowledge about the concepts and methods required for construction of large software intensive system.
2. To gain knowledge on the principles and techniques of Software Project Management.
3. To gain knowledge about organizational behavior and general Management techniques used for Project Management.

Course Outcomes:

After learning the course, the students should be able:

1. To apply the process to be followed in the software development life-cycle models.
2. To understand approaches for managing and optimizing the software development process.
3. To explain the quality management and different types of metrics used in software development.
4. To do the Project scheduling, Tracking, Risk Analysis, Quality Management and Project cost estimation using different techniques and tools.

Course Contents:

UNIT I

Project Evaluation and Planning - Activities in Software Project Management, Overview of Project Planning, Stepwise planning, Software processes and process models, Cost Benefit Analysis, Cash Flow Forecasting, Cost-Benefit Evaluation Techniques, Risk Evaluation.
Software effort estimation, Activity Planning, Risk Management, Resource Allocation.

UNIT II

Monitoring and Control- Collecting Data, Visualizing Progress, Cost Monitoring, Review techniques, Project termination review, Earned Value analysis, Change Control, Software Configuration Management (SCM).

UNIT III

Managing Contracts: Types of Contracts, Stages in Contract Placement, Typical Terms of a Contract, Contract Management and Acceptance.

UNIT IV

Quality Management and People Management- Introduction, Understanding Behavior, Organizational Behavior, Selecting The Right Person for the Job, Motivation, The Oldman – Hackman Job Characteristics Model, Working in Groups, Organization and team structures, Decision Making,

Leadership, Organizational Structures, Stress, Health and Safety. ISO and CMMI models, Testing, and Software reliability, Test automation.

UNIT V

Overview of Project Management Tools.

Text Book:

1. Bob Hughes, Mike Cotterell, ***"Software Project Management"***, Tata McGraw Hill, 6th Edition, 2017.

Reference Books:

1. Wakker Royce, ***"Software Project Management"***, Pearson Education, 2002.
2. Robert K. Wysocki, ***"Effective Software Project Management"***, Wiley, 2006.

Course Title:	**Introduction to Data Science**	**Semester**	**VI**
Course Code	**BTITOE604E**	**Course Type**	**Elective**
Prerequisite	**NIL**	**L – T – P**	**3 – 0 – 0**
Stream	**Open Elective**	**Credits**	**3**

Course Objectives:

1. To learn python data types (including list, tuple, set, and dictionary) and data cleaning methods.
2. To demonstrate an understanding of statistics and machine learning concepts that are vital for Data Science.
3. To learn to use data science libraries such asMatplotlib, NumPy, Scikit-learn, TensorFlow, Keras, Pandas etc. on the specific data set.

Course Outcomes:

After learning the course, the students should be able:

1. To understand Data Science Process.
2. To understand the mathematical foundations needed for Data Science.
3. To collect, explore, clean, munge and manipulate data.
4. To implement models such as linear regression, decisiontrees, and clustering.
5. To build Data Science applications using Python based toolkits.

Course Contents:

UNIT I

Introduction to Data Science: Concept of Data Science, Facets of Data,Overview of the data science process, Steps, Defining research goals and creating a project charter, Retrieving Data, Cleansing, integrating, and transforming data, Exploratory data analysis, Build the models, Presentation and automation. Data types, expressions, variables, and string operations; Data structures: lists and tuples and sets and dictionaries.

UNIT II

Mathematical Foundations:
Linear Algebra: Vectors, Matrices,Statistics: Describing a Single Set of Data, Correlation, Simpson's Paradox, Correlation and Causation. Probability and Distributions: Binomial and Poisson, Exponential, Normal and Cumulative Probability Distribution. Variance analysis. Hypothesis and Inference: Statistical Hypothesis Testing, Confidence Intervals, P-hacking, Bayesian Inference.

UNIT III

Machine Learning:
Linear Regression: Simple Linear Regression, Steps in Building a Regression Model, Building Simple Linear Regression Model, Model Diagnostics, Multiple Linear Regression, Classification Problems: Classification Overview, Binary Logistic Regression, Credit Classification, Gain Chart, and Lift Chart, Classification Tree (Decision Tree Learning), Gradient Descent Algorithm, Scikit-Learn Library for Machine Learning, Clustering, K-Means Clustering, Creating Product Segments Using Clustering,

Forecasting: Forecasting Overview, Components of Time-Series Data, Moving Average, Decomposing Time Series, Auto-Regressive Integrated Moving Average Models.

UNIT IV

Introduction to Programing tools for Data Science: Toolkits using Python: Matplotlib, NumPy, Scikit-learn, NLTK,TensorFlow,Keras,Pandas; Visualizing Data: Bar Charts, Line Charts, Scatter plots; Working with data: Reading Files, Dealing with Missing Values, Scraping the Web, Using APIs (Example: Using the Twitter APIs), Cleaning and Munging, Manipulating Data, Rescaling, Dimensionality Reduction.

UNIT V

Case Studies of Data Science Applications: Predicting malicious URLs, Building a recommender system inside a database, Weather forecasting, Stock market prediction, Object recognition.

Text Books:

1. Joel Grus, "***Data Science from Scratch: First Principles with Python***", O'Reilly Media,2015.
2. Davy Cielen, Arno D. B. Meysman, and Mohamed Ali, **"Introducing Data Science",** Dreamtech Press,2022.
3. Manaranjan Pradhan, U Dinesh Kumar, ***"Machine Learning using Python",*** Wiley India Pvt Ltd,2018.

Reference Books:

1. Aurélien Géron, "***Hands-On Machine Learning with Scikit-Learn and Tensor Flow:Concepts, Tools, and Techniques to Build Intelligent Systems***", 1st Edition, O'Reilly Media,2017.
2. Jain V.K., "***Data Sciences***", Khanna Publishing House, Delhi,2019.
3. Jain V.K., "***Big Data and Hadoop***", Khanna Publishing House, Delhi,2017.
4. Jeeva Jose, "***Machine Learning***", Khanna Publishing House, Delhi,2020.
5. Chopra Rajiv, "***Machine Learning***", Khanna Publishing House, Delhi,2018.

Course Title:	**Database Management Systems Lab**	**Semester**	**VI**
Course Code	**BTITL605**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 – 2**
Stream	**Core**	**Credit**	**1**

Lab Experiments List:

1. Creation of databases and use of SQL commands (DDL, DML and DCL).
2. Suitable exercises to practice SQL commands, may be given for Insert, Update and Delete.
3. Write SQL procedure for an application which uses exception handling.
4. Write SQL procedure for an application with cursors.
5. Write SQL for implementing Nested Queries.
6. Write SQL for implementing Join Queries.
7. Write a DBMS program to prepare reports for an application using functions.
8. Write SQL block containing triggers.
9. Write SQL block containing stored procedures.
10. Develop a menu driven, GUI-based database application in any one of the domains such as Banking, Billing, Library Management, Payroll, Insurance, Inventory, Healthcare etc. integrating all the features covered in the above exercises.

Course Title:	**Operating Systems Lab**	**Semester**	**VI**
Course Code	**BTITL606**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L – T – P**	**0– 0 – 2**
Stream	**Core**	**Credit**	**1**

Lab Experiments List:

1. Basics of UNIX commands
2. Shell Programming
3. Implement the following CPU scheduling algorithms:
   Round Robin, SJF, FCFS, Priority scheduling
4. Implement all file allocation strategies:
   Sequential, Indexed, Linked
5. Implement Semaphores
6. Implement all File Organization Techniques:
   Single level directory, Two level, Hierarchical, DAG
7. Implement Bankers Algorithm for Dead Lock Avoidance
8. Implement an Algorithm for Dead Lock Detection
9. Implement all page replacement algorithms:
   FIFO, LRU, LFU
10. Implement Shared memory and IPC
11. Implement Paging Technique of memory management
12. Implement Threading & Synchronization Applications

Course Title:	**Software Testing Lab**	**Semester**	**VI**
Course Code	**BTITL606**	**Course Type**	**Elective**
Prerequisite	**Software Engineering**	**L – T – P**	**0– 0 – 2**
Stream	**Software application & Development**	**Credit**	**1**

Lab Experiments List:

1. Design, develop, code and run the program in any suitable language to solve the commission problem, Analyze it from the perspective of data flow testing, derive at least 10 different test cases, execute these test cases and discuss the test results.

2. Design, develop, code and run the program in any suitable language to solve the Next Date problem, Analyze it from the perspective of decision table-based testing, derive at least 10 different test cases, execute these test cases and discuss the test results.

3. Design, develop, code and run the program in any suitable object-oriented language to solve the calendar problem. Analyze it from the perspective of OO testing, derive test cases to test the method that increment the date and the method that increments the month, execute these test cases and discuss the test results.

4. Design, develop, code and run the program in any suitable object-oriented language to solve the currency converter problem. Analyze it from the perspective of use case-based system testing, derive appropriate system test cases, execute these test cases and discuss the test results.

5. Design, develop, code and run the program in any suitable language to implement an absolute letter grading procedure, making suitable assumptions. Determine the basis paths and using them to derive different test cases, execute these test cases and discuss the test results.

6. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them to derive different test cases, execute these test cases and discuss the test results.

Course Title:	**Data Storage Technologies & Networks Lab**	**Semester**	**VI**
Course Code	**BTITL606**	**Course Type**	**Elective**
Prerequisite	**Computer Networks & Internetworking Protocol, Operating Systems**	**L – T – P**	**0 – 0 – 2**
Stream	**Infrastructure & Security Management**	**Credit**	**1**

Lab Experiments List:

1. Install a hard disk on a Linux machine covering all the below activities:

   a) Connecting the disk to an HBA (Host Bus Adapter) and BIOS setup for the disk;
   b) Partitioning the disk;
   c) Creating file systems within disk partitions;
   d) Mounting the files systems;
   e) Setting up automatic mounting;
   f) Labelling disk partitions;
   g) Setting up swapping on swap partitions.

2. Use "smartmontools" to monitor the disk performance monitoring and testing:

   a) Use "smartctl" to enable S.M.A.R.T. support and offline data collection on the disk;
   b) Check the overall health of the disk;
   c) Run a self-test on the disk;
   d) Set up "smartd" to do tests automatically.

3. Use "hdparm", "iostat", and "iometer" tools to measure the performance of different storage devices, such as SATA drive, SCSI drive, and USB drives.
   Plot graphs to compare read/write and sequential/random access rates among different storage devices.

4. Use Navisphere Manager Simulator to perform management on SAN disk array systems:
   a) Configure storage pools and LUNs (Logical Unit Number) for storage groups;
   b) Configure snapshots and clones;
   c) Create SANCopy full and incremental sessions;
   d) Create MirrorView synchronous and asynchronous images;
   e) Expand a LUN to create metaLUNs;
   f) Migrate a LUN to another LUN.

5. Use Openfiler for network storage configuration management:

   a) Configure the Openfiler to support locally attached USB drives;
   b) Set up a NAS server to support NSF and CIFS protocols;
   c) Set up a SAN server to support an iSCSI protocol.

6. Configure Openfiler as a NAS Server:

   a) Configure access control rules and NFS/CIFS shares for the NAS server;
   b) Configure the Linux client machine to access the NFS shares on the NAS server;
   c) Configure a Windows VM on the Linux client machine to access the CIFS shares on the NAS server;

d) Use Openfiler to set up a SAN server, to supports iSCSI protocol for the block level data access;
e) Configure access control rules for the SAN server and configure iSCSI targets on the server.

7. a) Use VMware to create virtual disks, Virtual Machine File Systems and provisioning;
b) Use thin and thick provisioning concepts.

Course Title:	**Service Oriented Architecture Lab**	**Semester**	**VI**
Course Code	**BTITL606**	**Course Type**	**Compulsory**
Prerequisite	**Nil**	**L - T - P**	**0- 0 - 2**
Stream	**Information Management & Quality Control**	**Credit**	**1**

Lab Experiments List:

1. To create a web service for adding a few numbers using NetBeans.
2. To create a web service for adding few numbers using NetBeans and write client-side code to invoke the web service.
3. Creation of a Web Service with Database Connectivity.
4. Create a SOA project with BPEL Module to compose a web service.
5. To develop a web service program which can persist the records of a student in the exam table. It makes use of SOAP Request and SOAP Response.
6. To invoke EJB components as web services.
7. To create a web services in .NET.
8. To invoke J2EE web services from .NET clients.
9. To create components using .NET client.
10. To access .NET web services from J2EE client.
11. Develop a Service Orchestration Engine (workflow) using WS-BPEL and implement service composition (Study Experiment).

Course Title:	**Network Programming Lab**	**Semester**	**VI**
Course Code	**BTITL606**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 – 2**
Stream	**Network**	**Credit**	**1**

Lab Experiments List:

1. Getting started with Basics of Network configurations files and Networking Commands in Linux.
2. To familiarize and understand the use and functioning of system calls used for operating system and network programming in Linux.
3. Familiarization and implementation of programs related to process and thread.
4. Implement programs for Inter Process Communication using PIPE, Message Queue and Shared Memory.
5. Implement Client-Server communication using Socket Programming and TCP as transport layer protocol.
6. Implement Client-Server communication using Socket Programming and UDP as transport layer protocol.
7. Implement and simulate algorithm for distance vector routing protocol.
8. Implement and simulate algorithm for link state routing protocol.
9. Implement Simple Mail Transfer Protocol.
10. Using Wireshark, observe data transferred in client server communication using UDP and identify the UDP datagram.
11. Using Wireshark, observe Three Way Handshaking Connection Establishment, Data Transfer and Three Way Handshaking Connection Termination in client server communication using TCP.
12. Develop a packet capturing and filtering application using raw sockets.

Course Title:	**Data Warehousing & Data Mining Lab**	**Semester**	**VI**
Course Code	**BTITL606**	**Course Type**	**Elective**
Prerequisite	**Nil**	**L – T – P**	**0 – 0 – 2**
Stream	**Data Science**	**Credit**	**1**

Lab Experiments List:

1. Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects,etc.).
2. Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, Manufacturing, Automobiles, Sales etc.).
3. Write ETL scripts and implement them using data warehouse tools.
4. Perform Various OLAP operations such as slice, dice, roll up, drill up and pivot.
5. Explore visualization features of the tool for analysis like identifying trends etc.
6. Explore WEKA Data Mining/Machine Learning Toolkit.
7. Load each dataset into Weka and run the Apriori algorithm with different support and confidence values. Study the rules generated.
8. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated. Derive interesting insights and observe the effect of discretization in the rule generation process.
9. Load each dataset into Weka and run ID3, J48 classification algorithm, study the classifier output. Compute entropy values, Kappa statistics.
10. Extract if-then rules from the decision tree generated by classifier, observe the confusion matrix and derive Accuracy, F- measure, TPrate, FPrate, Precision and recall values. Apply a cross-validation strategy with various fold levels and compare the accuracy results.
11. Load each dataset into Weka and run a simple k-means clustering algorithm with different values of k(number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.