



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

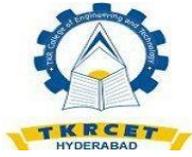
B.TECH COURSE STRUCTURE & SYLLABUS

V SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1	B65PC1	Operating Systems	3	0	0	3
3	B65PC2	Computer Networks	3	0	0	3
4	B65PC3	Data Warehousing & Data Mining	3	0	0	3
5	B65PC4	Design and Analysis of Algorithms	3	0	0	3
6	BHSFM	Fundamentals Of Management	3	0	0	3
7	B65PC6	Linux Programming	3	0	0	3
8	B65PC7	Operating Systems and Computer Networks lab	0	0	3	1.5
9	B65PC8	Linux Programming Lab	0	0	3	1.5
10	B65MC1	*Professional Ethics	0	3	0	0
Total Credits						21

VI SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1	B66PC1	Web Technologies	3	0	0	3
2	B66PC2	Object Oriented Analysis & Design	3	0	0	3
3	B66PC3	Compiler Design	3	0	0	3
4	B66PE4	1.Software Testing Methodologies 2.Software Project Management 3.Software Metrics and Quality Assurance	3	0	0	3
5	B66OE5	O.E – I	3	0	0	3
6	B66OE6	O.E – II	3	0	0	3
8	B66PC7	Web Technologies Lab	0	0	3	1.5
8	B66PC8	Case Tools Lab	0	0	3	1.5
9	B66MC2	*Environmental Science	0	3	0	0
Total Credits						21



TKR COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

OPERATING SYSTEMS - B65PC1

B.Tech V Semester

L/T/P/C

3/0/0/3

Course Objectives:

To gain insight knowledge on performance and working of an operation system.

Course Outcomes:

The student will be able to

1. Understand the importance of operating system.
2. Solve the synchronization problems.
3. Apply techniques involved in maximization of throughput by keeping CPU as busy as Possible.
4. Understand the role of memory management strategies and need of virtual memory.
5. Understand the mechanism for protecting the files by changing the access control.

UNIT – I:

Overview

Introduction-Operating system objectives, User view, System view, Operating system definition, Computer System Architecture, OS Structure, OS Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments. Operating System services, User and OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure.

UNIT – II:

CPU Scheduling Process

concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Operations on Processes, System calls-fork(),exec(),wait(),exit(), Inter-process communication-ordinary pipes and named pipes, message queues, shared memory, in Unix.

Process Scheduling

Basic concepts, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Linux scheduling and Windows scheduling. Process Synchronization, Background, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization in Linux and Windows.

UNIT – III:

Deadlocks

System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

Memory Management and Virtual Memory

Memory Management Strategies- Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of Page Table.

UNIT – IV:

Virtual Memory Management

Background, Demand Paging, Copy-on-Write, Page Replacement, Page Replacement Algorithms, Allocation of Frames, Thrashing, Virtual memory in Windows.Storage Management

File System- Concept of a File, System calls for file operations - open (), read (), write (), close (), seek (), unlink (), Access methods, Directory and Disk Structure, File System Mounting, File Sharing.

UNIT – V:

File System Implementation:

File System Structure, File System Implementation, Directory Implementation, Allocation methods, Free-space Management, Efficiency, and Performance. Overview of Mass Storage Structure.

Protection

System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

CASE STUDY on WINDOWS VISTA

History of windows vista, Programming windows vista, System structure, Process and threads in windows vista, memory management, Caching in windows vista, input/output in windows vista, the Windows NT file system, Security in windows vista.

TEXT BOOKS

1. Operating System Concepts , Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley, 2016 India Edition
2. Operating Systems – Internals and Design Principles, W. Stallings, 7th Edition, Pearson.

REFERENCE BOOKS

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI
2. Operating Systems: A concept-based Approach, 2nd Edition, D.M. Dhamdhare, TMH.
3. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
4. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
5. Principles of Operating systems, Naresh Chauhan, Oxford University Press.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

COMPUTER NETWORKS - B65PC2

B.Tech. V Semester

L/T/P/C

3/0/0/ 3

COURSE OBJECTIVES:

1. To introduce the fundamental various types of computer networks.
2. To demonstrate the TCP/IP and OSI models with merits and demerits.
3. To explore the various layers of OSI Model.
4. To introduce UDP and TCP Models.
5. To have the concept of different routing techniques for data communications.

COURSE OUTCOMES:

1. Students should understand and explore the basics of Computer Networks and Various Protocols. He/She will be in a position to understand the World Wide Web concepts.
2. Students will be in a position to administrate a network and flow of information further he/she can understand easily the concepts of network security, Mobile and ad hoc networks.

UNIT – I:

Introduction to Networks: Internet, Protocols and Standards, The OSI Model, Layers in OSI Model, TCP/IP Suite, Addressing. Physical Layer: Multiplexing, Transmission Media, Circuit Switched Networks, Datagram Networks, and Virtual Circuit Networks.

UNIT – II:

Data Link Layer: Introduction, Checksum, Framing, Flow and Error Control, Noiseless Channels, Noisy Channels, Random Access Controlled Access, Channelization, IEEE Standards, Ethernet, Giga-Bit Ethernet, Wireless LANs, SONET-SDH, Frame Relay and ATM.

UNIT – III:

Network Layer: Logical Addressing, Internetworking, Tunneling, Address Mapping, ICMP, IGMP, Forwarding, Routing-Flooding, Bellman& Ford, Disjkstra's routing protocols, RIP, OSPF, BGP,- and Multicast Routing Protocols. Connecting Devices-Passive Hubs, Repeaters, Active Hubs, Bridges, Routers.

UNIT – IV:

Transport Layer: Process to Process Delivery, UDP, TCP and SCTP Protocols, Congestion, Congestion Control, Quality of Service Application Layer: Domain Name Space, DNS in Internet, Electronic Mail, File Transfer Protocol, WWW, HTTP, SNMP, Multi-Media.

UNIT – V:

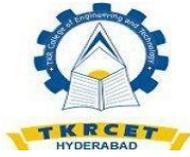
Network Security: Security services, mechanisms and attacks, IPsec, SSL, VPN, Firewall. Bluetooth, Zigbee, IPv4, IPv6.

TEXT BOOKS

1. Data Communications and Networking – Behrouz A. Forouzan, 4th Edition Mc Graw Hill Education, 2006.
2. Computer Networks — Andrew S Tanenbaum, 4th Edition, Pearson Education.
3. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose, K. W. Ross, 3rd Edition, Pearson Education.

REFERENCE BOOKS

1. Data communications and Networks by william stallings
2. Data communication and Networks – Bhusan Trivedi, Oxford university press 2016.
3. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
4. Understanding Communications and Networks, 3rd Edition, W.A.Shay, Cengage Learning



T K R COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

DATA WAREHOUSING & DATA MINING - B65PC3

B.Tech. V Semester

L/T/P/C
3/0/0/ 3

COURSE OBJECTIVE:

Study data warehouse principles and its working learn data mining concepts understand association rules mining. Discuss classification algorithms learn how data is grouped using clustering techniques.

COURSE OUTCOMES:

1. Students should be able to understand why the data warehouse in addition to database systems.
2. Ability to perform the preprocessing of data and apply mining techniques on it.
3. Ability to identify the association rules, classification and clusters in large data sets.
4. Ability to solve real world problems in business and scientific information using data mining.

UNIT – I:

Data Warehouse : Introduction to Data warehouse, Difference between operational database systems and data warehouses. Data warehouse Characteristics, Data warehouse Architecture and its Components, Extraction – Transformation – Loading, Logical (Multi – Dimensional), Data Modelling, Schema Design, Star and Snow – Flake Schema, Fact Consultation, Fact Table, Fully Addictive, Semi – Addictive, Non Addictive Measures; Fact Consultation, Fact Table, Fully Addictive, Semi – Addictive, Non Addictive Measures; Fact – Less – Facts, Dimension Table Characteristics; OLAP Cube, OLAP Operations, OLAP Server Architecture – ROLAP, MOLAP and HOLAP.

Introducing to Data Mining : Introduction to Data Mining : Introducing, What is Data Mining Difference between operational database systems and data warehouses, Data warehouses Characteristics, Data warehouse Architecture and its Components, Extraction – Transformation – Loading, Logical (Multi – Dimensional), Data Modeling, Schema Design, Star and Snow – Flake Schema, Fact Consultation, Fact Table, Fully Addictive, Semi – Addictive, Non Addictive Measures; Fact – Less – Facts, Dimension Table Characteristics; OLAP Cube, Olap Operations, OLAP Server Architecture – ROLAP, MOLAP and HOLAP.

UNIT – II:

Introducing to Data Mining : Introduction, What is Data Mining, Definition, KDD, Challenges, Data Mining Tasks, Data Preprocessing, Data Cleaning, Missing data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binaryzation, Data Transformation; Measures of Similarity and Dissimilarity – Basics.

UNIT – III:

Association Rules : problems Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation; APRIORI Algorithm, The Partition Algorithms, FP- Growth Algorithms, Compact Representation of Frequent Item set- Maximal Frequent Item Set, Closed Frequent Item Sets.

UNIT – IV:

Classification: Problem Definition, General Approaches to solving a classification problem, Evaluation of classifiers, Classification Techniques, Decision Tree – Decision tree Construction, Methods for Expressing attribute test conditions, Measures for Selecting the Best Split, Algorithm for Decision tree Induction; Naive Bayes Classifier, Bayesaian Belief Networks; K – N earnest neighbour classification – Algorithm and Characteristics.

UNIT – V:

Clustering: Problem Definition, Clustering Overview, Evaluation of Clustering Algorithms, Partitioning Clustering -K-Means Algorithm, K-Means Additional issues, PAM Algorithm; Hierarchical Clustering – Agglomerative Methods and divisive methods, Basic Agglomerative Hierarchical Clustering, Strengths ans Weakness; Outlier Detection.

TEXT BOOKS

1. Data Mining – Concepts and Techniques – Jiawei Han, Michelen Kamber, Morgan Kaufmann Publishers, Elsevier, 2 Edition, 2006.
2. Introduction to Data Mining, Pang – Ning Tan, Vipin Kumar, Michael Steinbach, Pearson Education.

REFERENCE BOOKS

1. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press.
2. Data Warehouse Fundamentals, Pualraj Ponnaiah, Wiley Student Edition.
3. Data Mining, Vikaram Pudi, P Radha Krishna, Oxford University Press



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

DESIGN AND ANALYSIS OF ALGORITHMS - B65PC4

B.Tech. V Semester

L/T/P/C
3/0/0/3

COURSE OBJECTIVES:

1. To analyze performance of algorithms.
2. To choose the appropriate data structure and algorithm design method for a specified application.
3. To understand how the choice of data structures and algorithm design methods impacts the performance of programs.
4. To solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking and branch and bound.
5. To understand the differences between tractable and intractable problems.
6. To introduce P and NP classes.

COURSE OUTCOMES:

1. Ability to analyze the performance of algorithms.
2. Ability to choose appropriate algorithm design techniques for solving problems.
3. Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs.

UNIT-I:

Introduction: What is an Algorithm, Algorithm specification, Performance analysis.

Divide and conquer- General method, applications - Binary search, Merge sort, Quick sort, Strassen's Matrix Multiplication.

UNIT – II:

Disjoint set operations, union and find algorithms, AND/OR graphs, Connected Components and Spanning trees, Bi-connected components **Backtracking**-General method, applications- The 8-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT – III:

Greedy method- General method, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

UNIT – IV:

Dynamic Programming- General Method, applications- Chained matrix multiplication, All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Traveling sales person problem.

UNIT – V:

Branch and Bound- General Method, applications-0/1 Knapsack problem, LC Branch and Bound solution, FIFO Branch and Bound solution, Traveling sales person problem.

NP-Hard and NP-Complete problems- Basic concepts, Non-deterministic algorithms, NP - Hard and NP- Complete classes, Cook's theorem.

TEXT BOOKS

1. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Sartaj Sahni and S. Rajasekharan, Universities Press.
2. Design and Analysis of Algorithms, P. H. Dave, H. B. Dave, 2nd edition, Pearson Education.

REFERENCE BOOKS

1. Algorithm Design: Foundations, Analysis and Internet examples, M. T. Goodrich and R. Tomassia, John Wiley and sons.
2. Design and Analysis of Algorithms, S. Sridhar, Oxford Univ. Press
3. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson Education.
4. Foundations of Algorithms,, R. Neapolitan and K. Naimipour, 4th edition, Jones and Bartlett Student edition.
5. Introduction to Algorithms, 3rd Edition, T. H. Cormen, C. E. Leiserson, R. L. Rivest, and Stein, PHI



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

FUNDAMENTALS OF MANAGEMENT - BHSFM

B. Tech: V Semester

L/T/P/ C
3/0/0/ 3

COURSE OBJECTIVE:

To understand the Management Concepts, applications of Concepts in Practical aspects of business and development of Managerial Skills.

COURSE OUTCOME:

The students understand the significance of Management in their Profession. The various Management Functions like Planning, Organizing, Staffing, Leading, Motivation and Control aspects are learnt in this course. The students can explore the Management Practices in their domain area.

UNIT – I:

Introduction to Management: Definition, Nature and Scope, Functions, Managerial Roles, Levels of Management, Managerial Skills, Challenges of Management; Evolution of Management- Classical Approach- Scientific and Administrative Management; The Behavioral approach; The Quantitative approach; The Systems Approach; Contingency Approach, IT Approach.

UNIT – II:

Planning and Decision Making: General Framework for Planning - Planning Process, Types of Plans, Management by Objectives; Development of Business Strategy. Decision making and Problem Solving - Programmed and Non Programmed Decisions, Steps in Problem Solving and Decision Making; Bounded Rationality and Influences on Decision Making; Group Problem Solving and Decision Making, Creativity and Innovation in Managerial Work.

UNIT – III:

Organization and HRM: Principles of Organization: Organizational Design & Organizational Structures; Departmentalization, Delegation; Empowerment, Centralization, Decentralization, Recentralization; Organizational Culture; Organizational Climate and Organizational Change. Human Resource Management & Business Strategy: Talent Management, Talent Management Models and Strategic Human Resource Planning; Recruitment and Selection; Training and Development; Performance Appraisal.

UNIT – IV:

Leading and Motivation: Leadership, Power and Authority, Leadership Styles; Behavioral Leadership, Situational Leadership, Leadership Skills, Leader as Mentor and Coach, Leadership during adversity and Crisis; Handling Employee and Customer Complaints, Team Leadership.

Motivation - Types of Motivation; Relationship between Motivation, Performance and Engagement, Content Motivational Theories - Needs Hierarchy Theory, Two Factor Theory, Theory X and Theory Y.

UNIT – V:

Controlling: Control, Types and Strategies for Control, Steps in Control Process, Budgetary and Non- Budgetary Controls. Characteristics of Effective Controls, Establishing control systems, Control frequency and Methods.

TEXT BOOKS

1. Management Fundamentals, Robert N Lussier, 5e, Cengage Learning, 2013.
2. Fundamentals of Management, Stephen P. Robbins, Pearson Education, 2009.

REFERENCES

1. Essentials of Management, Koontz Kleihrich, Tata McGraw Hill.
2. Management Essentials, Andrew DuBrin, 9e, Cengage Learning, 2012.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

LINUX PROGRAMMING - B65PC6

B. Tech: V SEM

**L/T/P/C
3/0/0/3**

COURSE OBJECTIVES:

1. To understand and make effective use of Linux utilities and Shell scripting language (bash) to solve Problems.
2. To implement in C some standard Linux utilities such as ls, mv, cp etc. using system calls.
3. To develop the skills necessary for systems programming including file system programming, process and signal management, and interprocess communication.
4. To develop the basic skills required to write network programs using Sockets.

COURSE OUTCOMES:

1. Work confidently in Linux environment.
2. Work with shell script to automate different tasks as Linux administration.

UNIT- I:

Linux Utilities - File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities.

Sed-Scripts, Operation, Addresses, Commands, Applications, awk-Execution, Fields and Records, Scripts, Operation, Patterns, Actions, Associative Arrays, String and Mathematical functions, System commands in awk, Applications.

Shell programming with Bourne again shell (bash) - Introduction, shell responsibilities, pipes and Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

UNIT- II:

Files and Directories - File Concept, File types, File System Structure, file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, creat, read, write, close, lseek, dup2, file status information-stat family, file and record locking- fcntl function, file permissions - chmod, fchmod, file ownership-chown, lchown, fchown, links-soft links and hard links – symlink, link, unlink.

Directories - Creating, removing and changing

Directories-mkdir, rmdir, chdir, obtaining current working directory-getcwd, Directory contents, Scanning Directories-opendir, readdir, closedir, rewinddir functions.

UNIT- III:

Process – Process concept, Layout of a C program image in main memory, Process environment-environment list, environment variables, getenv, setenv, Kernel support for process, process identification, process control - process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process, system call interface for process management-fork, vfork, exit, wait, waitpid, exec family, Process Groups, Sessions and Controlling Terminal, Differences between threads and processes.

Signals – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise, alarm, pause, abort, sleep functions.

UNIT- IV:

Interposes Communication - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes-creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs (Named pipes), differences between unnamed and named pipes, popen and pclose library functions.

Message Queues - Kernel support for messages, APIs for message queues, client/server example.

Semaphores - Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

UNIT- V:

Shared Memory - Kernel support for shared memory, APIs for shared memory, shared memory example. **Sockets** - Introduction to Berkeley Sockets, IPC over a network, Client- Server model, Socket address structures (Unix domain and Internet domain), Socket system calls for connection oriented protocol and connectionless protocol, example-client/server

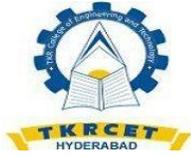
programs-Single Server-Client connection, Multiple simultaneous clients, Socket options set sock opt and fcntl system calls.

TEXT BOOKS

1. Unix System Programming using C++, T. Chan, PHI.
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
3. Unix Network Programming, W. R. Stevens, PHI.

REFERENCE BOOKS

1. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
2. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson.
3. System Programming with C and Unix, A. Hoover, Pearson.
4. Unix System Programming, Communication, Concurrency and Threads, K. A. Robbins and S. Robbins, Pearson Education.
5. Unix shell Programming, S. G. Kochan and P. Wood, 3rd edition, Pearson Education.
6. Shell Scripting, S. Parker, Wiley India Pvt. Ltd.
7. Advanced Programming in the Unix Environment, 2nd edition, W. R. Stevens and S. A. Rago, Pearson Education.
8. Unix and Shell programming, B. A. Forouzan and R. F. Gilberg, Cengage Learning.
9. Linux System Programming, Robert Love, O'Reilly, SPD.
10. C Programming Language, Kernighan and Ritchie, PHI



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

OPERATING SYSTEMS & COMPUTER NETWORKS LAB - B65PC7

B.Tech V Semester

L/T/P/C
0/0/3/1.5

COURSE OBJECTIVES:

1. To Understand the functionalities of various layers of OSI model
2. To understand the operating System functionalities

COURSE OUTCOMES:

1. Able to implement the various network security algorithms.
2. Able to implement and solve routing algorithms.
3. Able to implement various Scheduling algorithms.
4. Able to detect and solve deadlocks.
5. Able to implement Encryption algorithms.

System/ Software Requirement:

1. Intel based desktop PCs LAN CONNECTED with minimum of 166 MHZ or faster processor with atleast 64 MB
2. RAM and 100 MB free disk space.

COMPUTER NETWORKS LAB:

Part - A

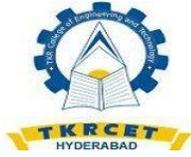
1. Implement the data link layer framing methods such as character, character stuffing and bit stuffing.
2. Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP .
3. Implement Dijkstra 's algorithm to compute the Shortest path thru a graph.
4. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table at each node using distance vector routing algorithm
5. Take an example subnet of hosts. Obtain broadcast tree for it.
6. Take a 64 bit playing text and encrypt the same using DES algorithm.
7. Write a program to break the above DES coding
8. Using RSA algorithm Encrypt a text data and Decrypt the same.

OPERATING SYSTEMS LAB:

Part -B

1. Simulate the following CPU scheduling algorithms
a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate all file allocation strategies
a) Sequential b) Indexed c) Linked
3. Simulate MVT and MFT
4. Simulate all File Organization Techniques
a) Single level directory b) Two level c) Hierarchical d) DAG
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention
7. Simulate all page replacement algorithms
a) FIFO b) LRU c) LFU Etc...

8. Simulate Paging Technique of memory management.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

LINUX PROGRAMMING LAB - B65PC8

B.Tech V Semester

L/T/P/C
0/0/3/1.5

COURSE OBJECTIVES:

1. To write shell scripts to solve problems.
2. To implement some standard Linux utilities such as ls,cp etc using system calls.
3. To develop network-based applications using C.

COURSE OUTCOMES:

1. Ability to understand the Linux environment
2. Ability to perform the file management and multiple tasks using shell scripts in Linux environment

List of sample problems:

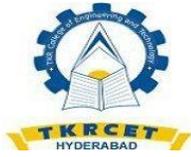
Note: Use Bash for Shell scripts.

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script to list all of the directory files in a directory.
7. Write a shell script to find factorial of a given integer.
8. Write an awk script to count the number of lines in a file that do not contain vowels.
9. Write an awk script to find the number of characters, words and lines in a file.
10. Write a C program that makes a copy of a file using standard I/O and system calls.
11. Implement in C the following Linux commands using System calls a) cat b) mv
12. Write a C program to list files in a directory.
13. Write a C program to emulate the Unix ls -l command.
14. Write a C program to list for every file in a directory, its inode number and file name.
15. Write a C program that redirects standard output to a file.Ex: ls > f1.
16. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
17. Write a C program to create a Zombie process.
18. Write a C program that illustrates how an orphan is created.
19. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex :- ls -l | sort
20. Write C programs that illustrate communication between two unrelated processes using named pipe (FIFO File).
21. Write a C program in which a parent writes a message to a pipe and the child reads the message.

22. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.
23. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (22)) and displays them.
24. Write a C program that illustrates suspending and resuming processes using signals.
25. Write Client and Server programs in C for connection oriented communication between Server and Client processes using Unix Domain sockets to perform the following:
Client process sends a message to the Server Process. The Server receives the message, reverses it and sends it back to the Client. The Client will then display the message to the standard output device.
26. Write Client and Server programs in C for connection oriented communication between Server and Client processes using Internet Domain sockets to perform the following:
Client process sends a message to the Server Process. The Server receives the message, reverses it and sends it back to the Client. The Client will then display the message to the standard output device.
27. Write C programs to perform the following: One process creates a shared memory segment and writes a message (“Hello”) into it. Another process opens the shared memory segment and reads the message (i.e. “Hello”). It will then display the message (“Hello”) to standard output device.

TEXT BOOKS

1. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
2. Advanced Unix Programming, N. B. Venkateswarulu, BS Publications.
3. Unix and Shell Programming, M.G. Venkatesh Murthy, Pearson Education.
4. Unix Shells by Example, 4th Edition, Ellie Quigley, Pearson Education.
5. Sed and Awk, O. Dougherty & A. Robbins, 2nd edition, SPD.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

PROFESSIONAL ETHICS - B65MC1

B.Tech. V Semester

L/T/P/C

0/3/0/0

COURSE OBJECTIVE:

To enable the students to imbibe and internalize the Values and Ethical Behaviour in the personal and Professional lives.

COURSE OUTCOME:

The students will understand the importance of Values and Ethics in their personal lives and professional careers. The students will learn the rights and responsibilities as an employee, team member and a global citizen.

UNIT – I:

Introduction to Professional Ethics: Basic Concepts, Governing Ethics, Personal & Professional Ethics, Ethical Dilemmas, Life Skills, Emotional Intelligence, Thoughts of Ethics, Value Education, Dimensions of Ethics, Profession and professionalism, Professional Associations, Professional Risks, Professional Accountabilities, Professional Success, Ethics and Profession.

UNIT – II:

Basic Theories: Basic Ethical Principles, Moral Developments, Deontology, Utilitarianism, Virtue Theory, Rights Theory, Casuist Theory, Moral Absolution, Moral Rationalism, Moral Pluralism, Ethical Egoism, Feminist Consequentialism, Moral Issues, Moral Dilemmas, Moral Autonomy.

UNIT – III:

Professional Practices in Engineering: Professions and Norms of Professional Conduct, Norms of Professional Conduct vs. Profession; Responsibilities, Obligations and Moral Values in Professional Ethics, Professional codes of ethics, the limits of predictability and responsibilities of the engineering profession.

Central Responsibilities of Engineers - The Centrality of Responsibilities of Professional Ethics; lessons from 1979 American Airlines DC-10 Crash and Kansas City Hyatt Regency Walk away Collapse.

UNIT – IV:

Work Place Rights & Responsibilities, Ethics in changing domains of Research, Engineers and Managers; Organizational Complaint Procedure, difference of Professional Judgment within the Nuclear Regulatory Commission (NRC), the Hanford Nuclear Reservation.

Ethics in changing domains of research - The US government wide definition of research misconduct, research misconduct distinguished from mistakes and errors, recent history of attention to research misconduct, the emerging emphasis on understanding and fostering responsible conduct, responsible authorship, reviewing & editing

UNIT – V:

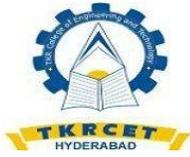
Global issues in Professional Ethics: Introduction – Current Scenario, Technology Globalization of MNCs, International Trade, World Summits, Issues, Business Ethics and Corporate Governance, Sustainable Development Ecosystem, Energy Concerns, Ozone Deflection, Pollution, Ethics in Manufacturing and Marketing, Media Ethics; War Ethics; Bio Ethics, Intellectual Property Rights.

TEXT BOOKS

1. Professional Ethics: R. Subramanian, Oxford University Press, 2015.
2. Ethics in Engineering Practice & Research, Caroline Whitbeck, 2e, Cambridge University Press 2015.

REFERENCE BOOKS

1. Engineering Ethics, Concepts Cases: Charles E Harris Jr., Michael S Pritchard, Michael J Rabins, 4e, Cengage learning, 2015.
2. Business Ethics concepts & Cases: Manuel G Velasquez, 6e, PHI, 2008.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18 WEB TECHNOLOGIES - B66PC1

B.Tech VI Semester

L/T/P/C
3/0/0/3

COURSE OBJECTIVES:

1. To introduce Client Side scripting with JavaScript and AJAX.
2. To introduce PHP language for Server Side Scripting.
3. To introduce XML and XML data with Java
4. To introduce server side programming with Java Servlets and JSP.

COURSE OUTCOMES:

1. Able to design a static web page using forms and frames with physical tags in html and validate client side scripting using onClick(),onSubmit(),onChange() events in javascript
2. Able to construct a validation page which connects to a data base given and able to perform the DML functionalites by using mysql_connect() ,mysql_query(), mysql_fetch_array(), mysql_close() in php
3. Students can analyze how to develop a wellformed and valid xml document by using DTDs and Schemas which allows the validation of text elements.
4. Able to write server side program by using servlets for given problem and able to develop a connection between both the ends by doGet() and doPost() methods.
5. Develop JSP applications implementing Session Management and Database Connectivity.

UNIT-I:

Introduction to HTML: HTML basic tags, list, table, image, forms, frames, Cascading style sheets.

Introduction to Java script: Java script language- declaring variables, scope of variables, functions, Java script objects, event handlers (on click, on submit etc.), Document Object Model. Simple AJAX application.

UNIT-II:

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like textboxes, radio buttons, list setc., Handling File Uploads, Connecting to database(My SQL as reference), executing simple queries, handling results, Handling sessions and cookies.

UNIT-III:

XML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, XHTML

Parsing XML Data- DOM and SAX Parsers in java.

UNIT-IV:

Introduction to Servlets: Common Gateway Interface (CGI), Lifecycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

UNIT– V:

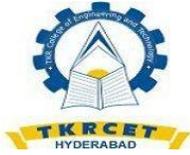
Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets ,implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

TEXT BOOKS

1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP–Steven Holzner, Tata McGraw-Hill.

REFERENCE BOOKS

1. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dreamtech
2. Java Server Pages– Hans Bergsten ,SPDO’ Reilly
3. JavaScript, D. Flanagan, O’Reilly, SPD.
4. Beginning Web Programming- Jon Duckett WROX.
5. Programming World Wide Web, R. W. Sebesta, Fourth Edition, Pearson.
6. Internet and World Wide Web– How to program, Dietel and Nieto, Pearson.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

OBJECT ORIENTED ANALYSIS AND DESIGN - B66PC2

B.Tech VI Semester

L/T/P/C
3/0/0/3

COURSE OBJECTIVES:

1. Concisely define the following key terms: class, object, state, behavior, object class, class diagram, object diagram, operation, encapsulation, constructor operation, query operation, update operation, scope operation, association, association role, multiplicity, association class, abstract class, concrete class, class-scope attribute, abstract operation, method, polymorphism, overriding, multiple classification, aggregation, and composition.
2. Describe the activities in the different phases of the object-oriented development life cycle.
3. State the advantages of object-oriented modeling vis-à-vis structured approaches.
4. Compare and contrast the object-oriented model with the E-R and EER models.
5. Model a real-world application by using a UML class diagram.
6. Provide a snapshot of the detailed state of a system at a point in time using a UML (Unified Modeling Language) object diagram.
7. Recognize when to use generalization, aggregation, and composition relationships.
8. Specify different types of business rules in a class diagram.

COURSE OUTCOMES:

Graduate can able to take up the case studies and model it in different views with respect user requirement such as use case, logical, component and deployment and etc, and preparation of document of the project for the unified Library application.

UNIT– I:

Introduction to UML: Importance of modelling, principles of modelling, object oriented modelling, conceptual model of the UML, Architecture, Software Development Life Cycle.

UNIT– II:

Basic Structural Modelling: Classes, Relationships, common Mechanisms, and diagrams. Advanced Structural Modelling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. Class & Object Diagrams: Terms, concepts, modelling techniques for Class & Object Diagrams.

UNIT– III:

Basic Behavioural Modelling-I: Interactions, Interaction diagrams. Basic Behavioural Modelling- II: Use cases, Use case Diagrams, Activity Diagrams.

UNIT– IV:

Advanced Behavioural Modelling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams Architectural Modelling: Component, Deployment, Component diagrams and Deployment diagrams.

UNIT– V:

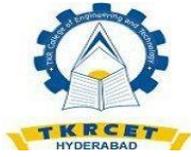
Patterns and Frameworks, Artificer Diagrams. Case Study: The Unified library application, **ATM application.**

TEXT BOOKS

1. Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education 2nd Edition
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

REFERENCE BOOKS

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML Pearson Education.
2. Pascal Rogues: Modeling Software Systems Using UML2, WILEY- Dream tech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McG raw Hills Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML, TMH.
5. Appling UML and Patterns: An introduction to Object — Oriented
6. Analysis and Design and Unified Process, Craig Larman, Pearson Education.
7. Object-Oriented Analysis and Design with the Unified Process by John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning.
8. UML and C++, R.C. Lee, and W.M. Tepfenhart, PHI.
9. Object Oriented Analysis, Design and Implementation, B. Dathan. S.Ramnath, Universities Press.
10. Design with UML and Java, K.Barclay, J.Savage, Elsevies.
11. Learning UML 2.0, Russ Miles and Kim Hamilton, O'Reilly, SPD.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

COMPILER DESIGN - B66PC3

B.Tech VI Semester

L/T/P/C
3/0/0/3

COURSE OBJECTIVES:

1. To understand the various phases in the design of a compiler.
2. To understand the design of top-down and bottom-up parsers.
3. To understand syntax directed translation schemes.
4. To introduce lex and yacc tools.
5. To learn to develop algorithms to generate code for a target machine.

COURSE OUTCOMES:

1. Ability to design, develop, and implement a compiler for any language.
2. Able to use lex and yacc tools for developing a scanner and a parser.
3. Able to design and implement LL and LR parsers.
4. Able to design algorithms to perform code optimization in order to improve the performance of a program in terms of space and time complexity.
5. Ability to design algorithms to generate machine code

UNIT – I:

Introduction: Language Processors, the structure of a compiler, the science of building a compiler, programming language basics.

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyzer Generator, Optimization of DFA-Based Pattern Matchers.

UNIT – II:

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers, Using Ambiguous Grammars, Parser Generators.

UNIT – III:

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax-Directed Translation, Syntax-Directed Translation Schemes, and Implementing L-Attributed SDD's.

Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code, Types and Declarations, Type Checking, Control Flow, Back patching, Switch-Statements, Intermediate Code for Procedures.

UNIT – IV:

Run-Time Environments: Storage organization, Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management, Introduction to Garbage Collection, Introduction to Trace-Based Collection.

Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, A Simple Code Generator, Peephole Optimization, Register Allocation and Assignment, Dynamic Programming Code-Generation.

UNIT – V:

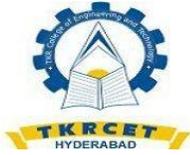
Machine-Independent Optimizations: The Principal Sources of Optimization, Introduction to Data-Flow Analysis, Foundations of Data-Flow Analysis, Constant Propagation, Partial Redundancy Elimination, Loops in Flow Graphs

TEXT BOOKS

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson.

REFERENCE BOOKS

1. Compiler Construction-Principles and Practice, Kenneth C Louden, Cengage Learning.
2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
3. The Theory and Practice of Compiler writing, J. P. Tremblay and P. G. Sorenson, TMH



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY-R18

SOFTWARE TESTING METHODOLOGIES - B66PE4

B. Tech VI Semester

L/T/P/C
3/0/0/3

COURSE OBJECTIVES:

To understand the software testing methodologies such as flow graphs and path testing, transaction flows testing, data flow testing, domain testing and logic base testing.

COURSE OUTCOMES:

1. Ability to apply the process of testing and various methodologies in testing for testing for developed software.
2. Ability to write test cases for given software to test it before delivery to the customer.

UNIT – I:

Introduction:- Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs. **Flow graphs and Path testing:-** Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT – II:

Transaction Flow Testing:- transaction flows, transaction flow testing techniques.

Dataflow testing:- Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

UNIT – III:

Domain Testing:- domains and paths, Nice & ugly domains, domain testing. domains and interfaces testing, domain and interface testing, domains and test ability.

UNIT-IV:

Paths, Path products and Regular expressions:- path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection. **Logic Based Testing:-** overview, decision tables, path expressions, kv charts, specifications.

UNIT – V:

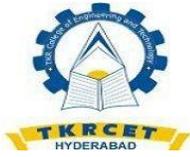
State, State Graphs and Transition testing:- state graphs, good & bad state graphs, state testing, Testability tips. **Graph Matrices and Application:-** Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

TEXT BOOKS

1. Software Testing techniques — Boris Beizer, Dreamtech, second edition.
2. Software Testing Tools — Dr.K.V.K.K.Prasad, Dreamtech.

REFERENCE BOOKS

1. The craft of software testing – Brian Marick, Pearson Education.
2. Software Testing, 3rd edition, P.C. Jorgensen, Aurbach publications (Dist.by SPD).
3. Software Testing, N. Chauhan, Oxford University press.
4. Introduction to Software Testing, P. Ammann & Joffutt, Cambridge Univ. Press.
5. Effective methods of Software Testing, Perry, John Wiley, 2nd Edition, 1999.
6. Software Testing Concepts and Tools, P.Nageswara Rao, Dreamtech Press.
7. Software Testing, M.G.Limaye, TMH.
8. Software Testing, S.Desikan, G.Ramesh, Pearson.
9. Foundations of Software Testing, D.Graham & other, Cengage Learning.
10. Foundations of Software Testing, A.P.Mathur, Pearson.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18 SOFTWARE PROJECT MANAGEMENT - B66PE4

B.Tech. VI Semester

L/T/P/C
3/0/0/3

COURSE OBJECTIVES:

The main goal of software development projects is to create a software system with a predetermined functionality and quality in a given time frame and with given costs. For achieving this goal, models are required for determining target values and for continuously controlling these values. This course focuses on principles, techniques, methods & tools for model-based management of software projects. Assurance of product quality and process adherence (quality assurance), as well as experience-based creation & improvement of models (process management). The goals of the course can be characterized as follows.

1. Understanding the specific roles within a software organization as related to project and process management
2. Understanding the basic infrastructure competences (e.g., process modeling and measurement)
3. Understanding the basic steps of project planning, project management, Quality assurance, and process management and their relationships

COURSE OUTCOMES:

1. Describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project.
2. Compare and differentiate organization structures and project structures.
3. Implement a project to manage project schedule, expenses and resources with the application of suitable project management tools.

UNIT- I:

Conventional Software Management: The waterfall model, conventional software Management performance. Evolution of Software Economics: Software Economics. Pragmatic software cost estimation.

UNIT- II:

Improving Software Economics: Reducing Software product size, Improving software processes, improving team effectiveness. Improving automation, Achieving required quality, peer inspections. The old way and the new- The principles of conventional software engineering. Principles of modern software management, transitioning to an iterative process.

UNIT- III:

Life cycle phases: Engineering and production stages, inception. Elaboration, construction, transition phases. Artifacts of the process: The artifact sets. Management artifacts, Engineering artifacts, programmatic artifacts. Model based software architectures: A Management perspective and technical perspective.

UNIT- IV:

Work Flows of the process: Software process workflow, Inter trans workflows. Checkpoints of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments. Iterative Process Planning Work breakdown structures, planning guidelines, cost and scheduled estimating,

Interaction, planning process, Pragmatic planning.

1. Project Organizations and Responsibilities: Line-of-Business Organizations,
2. Project Organizations, evolution of Organizations.
3. Process Automation: Automation Budding Blocks. The Project Environment..

UNIT-V:

Project Control and Process instrumentation: The server care Metrics, Management indicators, and quality indicators. life cycle expectations pragmatic Software Metrics, Metrics automation. Tailoring the Process: Process discriminates, Example. Future Software Project Management: Modern Project Profiles Next generation Software economics modern Process transitions.

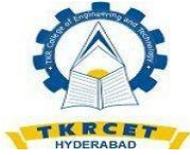
Case Study: The Command Center Processing and Display System. Replacement (CCPDS. R).

TEXT BOOKS

1. Software Project Management. Walker Royce, Pearson Education.
2. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, Tate McGraw Hd.

REFERENCE BOOKS

1. Applied Software Project Management, Andrew SteIbian & Jennifer Greene, O'Reilly. 2006
2. Head First PMP, Jennifer Greene & Andrew Steliman, O'RoiHy.2007
3. Software Enneeñing Project Managent. Richard H. Thayer & Edward Yourdon, second edition, Wiley India, 2004.
4. Ale Project Management, Jim Highsniith. Pearson education, 2004
5. The art of Project management. Scott Berkun. O'Reilly, 2005.
6. Software Project Management in Practice. Pankaj Jalote. Pearson Educabon,2002.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

SOFTWARE METRICS AND QUALITY ASSURANCE - B66PE4

B.Tech. VI Semester

**L/TP/C
3/0/0/3**

COURSE OBJECTIVES:

1. To gain basic knowledge about metrics, measurement theory and related terminologies
2. To learn measure the quality level of internal and external attributes of the software product
3. To introduce the basics of software reliability and to illustrate how to perform planning, executing and testing for software reliability
4. To explore various metrics and models of software reliability
5. To compare various models of software reliability based on its application

COURSE OUTCOMES:

Upon completion of the course, students shall be able to

1. Identify and apply various software metrics, which determines the quality level of software
2. Identify and evaluate the quality level of internal and external attributes of the software product
3. Compare and Pick out the right reliability model for evaluating the software
4. Evaluate the reliability of any given software product
5. Design new metrics and reliability models for evaluating the quality level of the software based on the requirement.

UNIT-I:

What Is Software Quality: Quality: Popular Views, Quality Professional Views, Software Quality, Total Quality Management, and Summary. Fundamentals Of Measurement Theory: Definition, Operational Definition, And Measurement, Level Of Measurement, Some Basic Measures, Reliability And Validity, Measurement Errors, Be Careful With Correlation, Criteria For Causality, Summary. Software Quality Metrics Overview: Product Quality Metrics, In Process Quality Metrics, Metrics for Software Maintenance, Examples for Metrics Programs, Collecting software Engineering Data.

UNIT-II:

Applying The Seven Basic Quality Tools In Software Development: Ishikawa's Seven Basic Tools, Checklist, Pareo Diagram, Histogram, Run Charts, Scatter Diagram, Control Chart, Cause, and Effect Diagram. The Rayleigh Model: Reliability Models, the Rayleigh Model Basic Assumptions, Implementation, Reliability and Predictive Validity.

UNIT-III:

Complexity Metrics and Models: Lines of Code, Halstead's Software Science, Cyclomatic Complexity Syntactic Metrics, An Example of Module Design Metrics in Practice .Metric And Lessons Learned for Object Oriented Projects: Object Oriented Concepts And Constructs, Design And Complexity Metrics, Productivity Metrics, Quality And Quality Management Metrics, Lessons Learned For object oriented Projects.

UNIT-IV:

Availability Metrics: Definition and Measurement of System Availability, Reliability Availability and Defect Rate, Collecting Customer Outage Data For Quality Improvement, In Process Metrics For Outage And Availability. Conducting Software Project Assessment: Audit Ad Assessment, Software Process Maturity Assessment And Software Project Assessment, Software Process Assessment A Proponed Software Project Assessment Method.

UNIT-V:

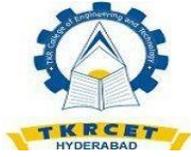
Dos And Don'ts Of Software Process Improvement :Measuring Process Maturity, Measuring Process Capability, Staged Versus Continuous Debating Religion, Measuring Levels Is Not Enough, Establishing The Alignment Principle ,Take Time Getting Faster, Keep it Simple Or Face Decomplexification, Measuring The Value Of Process Improvement ,Measuring Process Compliance , Celebrate The Journey Not Just The Destination. Using Function Point Metrics to Measure Software Process Improvement: Software Process Improvement Sequences, Process Improvement Economies, Measuring Process Improvement at Activity Levels.

TEXT BOOKS

1. Norman E-Fentor and Share Lawrence Pflieger.” Software Metrics”. International Thomson Computer Press, 1997.
2. Stephen H Khan: Metrics and Models in Software Quality Engineering, Pearson 2nd edition 2013.

REFERENCE BOOKS

1. S.A. Kelkar, “Software quality and Testing, PHI Learning, Pvt., Ltd., New Delhi 2012.
2. Watts S Humphrey, “Managing the Software Process”, Pearson Education Inc, 2008.
3. Mary Beth Chrissis, Mike Konrad and Sandy Shrum, “CMMI”, Pearson Education (Singapore) Pvt. Ltd., 2003
4. Philip B Crosby,” Quality is Free: The Art of Making Quality Certain “, Mass Market, 1992.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18 WEB TECHNOLOGIES LAB - B66PC7

B.Tech. VI Semester

**L/T/P/C
0/0/3/1.5**

COURSE OBJECTIVES:

To enable the student to program web applications using the following technologies HTML, Java Script, AJAX, PHP, Tomcat Server, Servlets, JSP.

COURSE OUTCOMES:

1. Able to create the web pages using HTML tags
2. Able to implement various tags.
3. Able to create web page using PHP.
4. Ability to establish database connectivity using Servlets.
5. Ability to use AJAX to modify the existing page.

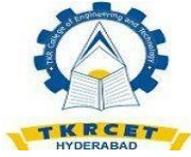
Note:

1. Use LAMP Stack (Linux, Apache, MySQL and PHP) for the Lab Experiments. Though not mandatory, encourage the use of Eclipse platform wherever applicable
2. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed
1. Install the following on the local machine Apache Web Server (if not installed) Tomcat Application Server locally
Install MySQL (if not installed)
Install PHP and configure it to work with Apache web server and MySQL (if not already configured)
2. Write an HTML page including java script that takes a given set of integer numbers and shows them after sorting in descending order.
3. Write an HTML page including any required Java script that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show “out of range” and if it is not a number, it should show “not a number” message in the result box.
4. Write a n HTML page that has one input, which can take multi-line text and a submit button. Once the user clicks the submit button it should show the number of characters, words and lines in the text entered using an alert message. Words are separated with white space and lines are separated with new line character.
5. Write an HTML page that contains a selection box with a list of 5countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and fontsize).
6. Create an XML document that contains 10users information. Write a Java program, which takes User Id as input and returns the user details by taking the user information from the XML document using (a) DOM Parser and (b) SAX parser
Implement the following web applications using (a) PHP (b) Servlets and (c) JSP:
7. A user validation web application, where the user submits the login name and password to the server. The name and password are checked against the data already available in Database and if the data matches, a successful login page is returned. Otherwise a failure message is shown to the user.
8. Modify the above program to use an XMLfile instead of database.

9. Modify the above program to use AJAX to show the result on the same page below the submit button.
10. A simple calculator web application that takes two numbers and an operator (+,-,/,*and %) from an HTML page and returns the result page with the operation performed on the operands.
11. Modify the above program such that it stores each query in a data base and checks the data base first for the result. If the query is already available in the DB, it returns the value that was previously computed (fromDB) or it computes the result and returns it after storing the new query and result in DB.
12. A web application takes a name as input and on submit it shows a hello <name> page where <name> is taken from the request. It shows the start time at the right top corner of the page and provides a logout button. On clicking this button, it should show a logout page with ThankYou <name> message with the duration of usage (hint:Use session to store name and time).
13. A web application that takes name and age from an HTML page. If the age is less than 18 , it should send a page with “Hello <name> ,you are not authorized to visit this site” message, where <name> should be replaced with the entered name. Otherwise it should send“ Welcome <name> to this site” message.
14. A web application for implementation:
The user is first served a login page which takes user's name and password. After submitting the details the server checks the values against the data from a database and takes the following decisions.
If name and password matches, serves a welcome page with user's full name.
If name matches and password doesn't match, then serves “password mis match” page
If name is not found in the database, serves a registration page, where user’s full name is asked and on submitting the full name, it stores, the login name, password and full name in the data base (hint:use session for storing the submitted login name and password)
15. A web application that lists all cookies stored in the browser on clicking “List Cookies” button. Add cookies if necessary.

REFERENCE BOOKS

1. The Complete Reference PHP– Steven Holzner, Tata McGraw-Hill
2. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dream tech
3. Java Server Pages–Hans Bergsten, SPDO’ Reilly
4. Java Script, D. Flanagan, O’ Reilly, SPD.
5. Internet and World Wide Web–How to program, Dieteland Nieto, Pearson.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18

CASE TOOLS LAB - B66PC8

B.Tech VI Semester

L/T/P/C

0/0/3/1.5

COURSE OBJECTIVES:

1. Understand how UML supports the entire OQAD process.
2. Become familiar with all phases of OQAD.

COURSE OUTCOMES:

1. Ability to understand the history, cost of using and building CASE tools.
2. Ability to construct and evaluate hybrid CASE tools by integrating existing tools.
3. Ability to deliver the product with qualitative.

CASE TOOLS LAB

Students are divided into batches of 5 each and each batch has to draw the following diagrams using UML for an ATM system whose description is given below. UML diagrams to be developed are:

- 1) Use Case Diagram.
- 2) Class Diagram.
- 3) Sequence Diagram.
- 4) Collaboration Diagram.
- 5) State Diagram
- 6) Activity Diagram.
- 7) Component Diagram
- 8) Deployment Diagram.
- 9) Test Design.
- 10) Description for an ATM System

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.) The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) – both of which will be sent to the bank for validation as part of each transaction.

The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned – except as noted below.

The ATM must be able to provide the following services to the customer

- 1) A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the bank before cash is dispensed.
- 2) A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
- 3) A customer must be able to make a transfer of money between any two accounts linked to the

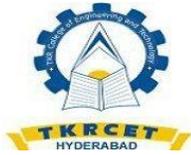
card.

- 4) A customer must be able to make a balance inquiry of any account linked to the card.
- 5) A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.) If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back. If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction. The ATM will provide the customer with a printed receipt for each successful transaction.

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the non"

Position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.



T K R COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY -R18 ENVIRONMENTAL SCIENCE - B66MC2

B. Tech VI Semester

L/T/P/C
0/3/0/0

COURSE OBJECTIVES:

1. Understanding the importance of ecological balance for sustainable development.
2. Understanding the impacts of developmental activities and mitigation measures.
3. Understanding the environmental policies and regulations

COURSE OUTCOMES:

Based on this course, the Engineering graduate will understand /evaluate / develop technologies on the basis of ecological principles and environmental regulations which in turn helps in sustainable development

UNIT-I:

Ecosystems: Definition, Scope, and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Bio magnifications, ecosystem value, services and carrying capacity, Field visits.

UNIT-II:

Natural Resources: Classification of Resources: Living and Non-Living resources, water resources: use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources, Land resources: Forest resources, Energy resources: growing energy needs, renewable and non renewable energy sources, use of alternate energy source, case studies.

UNIT-III:

Biodiversity And Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation. National Biodiversity act.

UNIT-IV:

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, Air Pollution: Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. Water pollution: Sources and types of pollution, drinking water quality standards. Soil Pollution: Sources and types, Impacts of modern agriculture, degradation of soil. Noise Pollution: Sources and Health hazards, standards, Solid waste: Municipal Solid Waste management, composition and characteristics of e-Waste and its management. Pollution control technologies: Wastewater Treatment methods: Primary, secondary and Tertiary. Overview of air pollution control technologies, Concepts of bioremediation. Global Environmental Issues and Global Efforts: Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol. NAPCC-GoI Initiatives.

UNIT-V:

Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act-1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules.

EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio-economical aspects. Strategies for risk assessment, Concepts of Environmental Management Plan (EMP). Towards Sustainable Future: Concept of Sustainable Development Goals, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

TEXT BOOKS

1. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
2. Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS

1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHLLearning Private Ltd. New Delhi.
2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.
3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
5. Introduction to Environmental Science by Y. Anjaneyulu, BS.Publications.