DISTRIBUTED SYSTEMS

Objectives:
- To understand what and why a distributed system is.
- To understand theoretical concepts, namely, virtual time, agreement and consensus protocols.
- To understand the DFS and DSM Concepts.
- To understand the concepts of transaction in distributed environment and associated concepts, namely, concurrency control, deadlocks and error recovery.

UNIT-I


UNIT-II
Time and Global States: Introduction, Clocks Events and Process States, Synchronizing Physical Clocks, Logical Time and Logical Clocks, Global States, Distributed Debugging.

Coordination and Agreement: Introduction, Distributed Mutual Exclusion, Elections, Multicast Communication, Consensus and Related Problems.

UNIT-III
InterProcess Communication: Introduction, The API for the Internet Protocols, External Data Representation and Marshalling, Client-Server Communication, Group Communication, Case Study: RPC in UNIX.

Distributed Objects and Remote Invocation: Introduction, Communication between Distributed Objects, Remote Procedure Call, Events and Notifications, Case Study: JAVA RMI.

UNIT-IV

Name Service: Introduction, Name Services and the Domain Name System, Directory Services, Case Study of the Global Name Services.

Distributed Shared Memory: Introduction, Design and Implementation Issues, Sequential Consistency and IVY case study, Release Consistency.
Objectives:
- Explain the objectives of information security
- Explain the importance and application of each of confidentiality, integrity, authentication and availability
- Understand various cryptographic algorithms.
- Understand the basic categories of threats to computers and networks
- Describe public-key cryptosystem.
- Describe the enhancements made to IPv4 by IPv6
- Understand Intrusions and intrusion detection
- Discuss the fundamental ideas of public-key cryptography.
- Generate and distribute a PGP key pair and use the PGP package to send an encrypted e-mail message.
- Discuss Web security and Firewalls

UNIT - I

Cryptography: Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition technique, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

UNIT - II
Symmetric key Cryptos: Block Cipher principles & Algorithms (DES, AES, Blowfish), Differential and Linear Cryptanalysis, Block cipher modes of operation, Stream ciphers, RLC4, Location and placement of encryption function, Key distribution: Asymmetric key Cryptos: Principles of public key cryptosystems, Algorithms (RSA, Diffie-Hellman, ECC), Key Distribution.

UNIT - III
II Year B.Tech. CSE-II Sem
L T P/D C
4 4 - - 4

(AE0534) OBJECT ORIENTED ANALYSIS AND DESIGN

Objectives:
- 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.
- Describe the activities in the different phases of the object-oriented development life cycle.
- State the advantages of object-oriented modeling vis-a-vis structured approaches.
- Compare and contrast the object-oriented model with the E-R and EER models.
- Model a real-world application by using a UML class diagram.
- Provide a snapshot of the detailed state of a system at a point in time using a UML (Unified Modeling Language) object diagram.
- Recognize when to use generalization, aggregation, and composition relationships.
- Specify different types of business rules in a class diagram.

UNIT-I
Introduction to UML: Importance of modeling, principles of modeling, object-oriented modeling, conceptual model of the UML, Architecture, Software Development Lifecycle.

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, modeling techniques for Class & Object Diagrams.

UNIT-III
Basic Behavioral Modelling-II: Use cases, Use case diagrams, Activity diagrams.
SOFTWARE TESTING METHODOLOGIES

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.

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 testability.

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, k-v 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:

REFERENCE BOOKS:
1. The craft of software testing - Brian Marick, Pearson Education.
7. Software Testing, M.G.Limaye, TMH.

Outcomes:
• Ability to apply the process of testing and various methodologies in testing for developed software.
• Ability to write test cases for given software to test it before delivery to the customer.
Unit V


TEXT BOOKS:

REFERENCES:

Outcomes:
At the end of the course, the student will:
* Understand the market dynamics namely, demand and supply, demand forecasting, elasticity of demand and supply, pricing methods and pricing in different market structures,

Unit IV


Unit III


Unit II


Unit I


- Gain an insight into how production function is carried out to achieve least cost combination of inputs and cost analysis
- Develop an understanding of
- Analyse how capital budgeting decisions are carried out
- Understand the framework for both manual and computerised accounting process
- Know how to analyse and interpret the financial statements through ratio analysis.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

III Year B.Tech. CSE-II Sem

L   T/P/D   C
4    4     4

(A60512) WEB TECHNOLOGIES

Objectives:
- To introduce PHP language for server side scripting
- To introduce XML and processing of XML, Data with Java
- To introduce Server side programming with Java Servlets and JSP
- To introduce Client side scripting with Javascript and AJAX.

UNIT I
Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads, Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies

FILE HANDLING IN PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories

UNIT II
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 III
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 IV
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.

UNIT V
Client side Scripting: Introduction to JavaScript: Javascript language - declaring variables, scope of variables, functions, event handlers (onclick, onsubmit etc.), Document Object Model, Form validation, Simple AJAX application.
TEXT BOOKS:
1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill

REFERENCE BOOKS:
2. Java Server Pages – Hans Bergsten, SPD O'Reilly
3. Java Script, D.Parnas, O'Reilly, SPD.
4. Beginning Web Programming- Jon Duckett WPDX.
6. Internet and World Wide Web – How to program, Dietel and Nieto, Pearson.

Outcomes:
• gain knowledge of client side scripting, validation of forms and AJAX programming
• have understanding of server side scripting with PHP language
• have understanding of what is XML and how to parse and use XML Data with Java
• To introduce Server side programming with Java Servlets and JSP

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
III Year B.Tech. CSE-II Sem
L T P D C

(60591) CASE TOOLS and WEB TECHNOLOGIES LAB

CASE TOOLS LAB

Objectives:
• Understand how UML supports the entire OOAD process.
• Become familiar with all phases of OOAD.
• Understand different software testing tools and their features

1. 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.
5. State Diagram
6. Activity Diagram.
7. Component Diagram
8. Deployment Diagram.

Description for an ATM System
The software to be designed will control a simulated automated teller machine (ATM) having a magnetic strip 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 serve 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 into 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 ATM 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 "on" 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 relasoe the machine with cash, blank receipts, etc.

II. Study of any testing tool (e.g. Win runner)
III. Study of any web testing tool (e.g. Selenium)
IV. Study of any bug tracking tool (e.g. Bugzilla, Jira)
V. Study of any test management tool (e.g. Test Director)
VI. Study of any open source-testing tool (e.g. Test Link)

Outcomes:

- Ability to understand the history, cost of using and building CASE tools.
- Ability to construct and evaluate hybrid CASE tools by integrating existing tools.

WEB TECHNOLOGIES LAB

Objectives:

- To enable the student to program web applications using the following technologies HTML, JavaScript, AJAX, PHP, Tomcat Server, Servlets, JSP

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 any required JavaScript that takes a number from one text field in the range of 0 to 999 and displays 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.

3. Write an 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.

4. Write an HTML page that contains a selection box with a list of 5 countries. 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 font size).

5. Create an XML document that contains 10 users 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.

6. Implement the following web applications using (a) PHP, (b) Servlets and (c) JSP.

i. 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.

ii. Modify the above program to use an xml file instead of database.

iii. Modify the above program to use AJAX to show the result on the same page below the submit button.

iv. A simple calculator web application that takes two numbers and an operator (+, -, *, /) from an HTML page and returns the result page with the operation performed on operands.

v. Modify the above program such that it stores each query in a database and checks the database first for the result. If the query is already available in the DB, it returns the value that was previously computed from DB or it computes the result and returns it after storing the new query and result in DB.

vi. 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 Thank You <name> message with the duration of usage (hint: use session to store name and time).

vii. 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 a "Welcome <name> to this site" message.

viii. 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 these values against the data from a database and takes the following occasions.

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 mismatch" 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 database (hint: use session for storing the submitted login name and password).

ix. A web application that lists all cookies stored in the browser on clicking "List Cookies" button. Add cookies if necessary.

TEXT BOOKS:
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill

REFERENCE BOOKS:
2. Java Server Pages – Hans Bergsten, SPD O'Reilly
3. Java Script, D.Flanagan, O'Reilly, SPD.
4. Beginning Web Programming–Jon Duckett WROX.
6. Internet and World Wide Web – How to program, Dietel and Nitesh, Pearson.

Outcomes:
1. Use LAMP Stack for web applications
2. Use Tomcat Server for Servlets and JSPs.
3. Write simple applications with Technologies like HTML, Javascript, AJAX, PHP, Servlets and JSPs
4. Connect to Database and get results.
5. Parse XML files using Java (DOM and SAX parsers)
The introduction of the Advanced Communication Skills Lab is considered essential at 3rd year level. At this stage, the students need to prepare themselves for their careers which may require them to listen to, read, speak and write in English both for their professional and interpersonal communication in the globalised context.

The proposed course should be a laboratory course to enable students to use ‘good’ English and perform the following:

- Gathering ideas and information to organise ideas relevantly and coherently.
- Engaging in debates.
- Participating in group discussions.
- Facing interviews.
- Writing project/research reports/technical reports.
- Making oral presentations.
- Writing formal letters.
- Transferring information from non-verbal to verbal texts and vice-versa.
- Taking part in social and professional communication.

Objectives:

This Lab focuses on using multi-media instruction for language development to meet the following targets:

- To improve the students’ fluency in English through a well-developed vocabulary and enable them to listen to English spoken at normal conversational speed by educated English speakers and respond appropriately in different socio-cultural and professional contexts.
- Further, they would be required to communicate their ideas relevantly and coherently in writing.
- To prepare all the students for their placements.

Syllabus:

The following course content to conduct the activities is prescribed for the Advanced Communication Skills (ACS) Lab:

1. Activities on Fundamentals of Inter-personal Communication and Building Vocabulary - Starting a conversation – responding appropriately and relevantly – using the right body language – Role Play in different situations & Discourse Skills - using visuals - Synonyms and antonyms, word roots, one-word substitutes, prefixes and suffixes, study of word origin, business vocabulary, analogy, idioms and phrases, colloquialisms & usage of vocabulary.
2. Activities on Reading Comprehension – General Vs Local comprehension, reading for facts, guessing meanings from context, scanning, skimming, inferring meaning, critical reading & effective googling.
4. Activities on Presentation Skills – Oral presentations (individual and group) through JAM sessions/seminars/PETs and written presentations through posters/projects/reports/ e-mails/assignments etc.
5. Activities on Group Discussion and Interview Skills – Dynamics of group discussion, intervention, summating, modulation of voice, body language, relevance, fluency and organization of ideas and coherence for evaluation - Concept and process, pre-interview planning, opening strategies, answering strategies, interview through tele-conference & video-conference and Mock Interviews.

Minimum Requirement:

The Advanced Communication Skills (ACS) Laboratory shall have the following infra-structural facilities to accommodate at least 35 students in the lab:

- Spacious room with appropriate acoustics.
- Round Tables with movable chairs
- Audio-visual aids
- LCD Projector
- Public Address system
- P - IV Processor, Hard Disk – 80 GB, RAM-512 MB Minimum, Speed – 2.8 GHZ
- T.V. a digital stereo & Camcorder
- Headphones of high quality
