JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem. (CSE)

(13A05401) COMPUTER ORGANIZATION AND ARCHITECTURE

Course Objective:

- *To learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design*
- *To make the students understand the structure and behavior of various functional modules of a computer.*
- *To understand the techniques that computers use to communicate with I/O devices*
- *To study the concepts of pipelining and the way it can speed up processing.*
- *To understand the basic characteristics of multiprocessors*

Learning Outcome:

- *Ability to use memory and I/O devices effectively*
- *Able to explore the hardware requirements for cache memory and virtual memory*
- *Ability to design algorithms to exploit pipelining and multiprocessors*

UNIT I

Introduction to Computer Organization and Architecture

Basic Computer Organization – CPU Organization – Memory Subsystem Organization and Interfacing – I/O Subsystem Organization and Interfacing – A Simple Computer Levels of Programming Languages, Assembly Language Instructions, Instruction Set Architecture Design, A simple Instruction Set Architecture

UNIT II

CPU Design and Computer Arithmetic

CPU Design: Instruction Cycle – Memory – Reference Instructions – Input/output and Interrupt – Addressing Modes – Data Transfer and Manipulation – Program Control.

Computer Arithmetic: Addition and Subtraction – Multiplication Algorithms – Division Algorithms – Floating-Point Arithmetic Operations – Decimal Arithmetic unit.

UNIT III

Register Transfer Language and Design of Control Unit

Register Transfer: Register Transfer Language – Register Transfer – Bus and Memory Transfers – Arithmetic Micro operations – Logic Micro operations – Shift Micro operations.

Control Unit: Control Memory – Address Sequencing – Micro program Example – Design of Control Unit.

UNIT IV

Memory and Input/output Organization

Memory Organization: Memory Hierarchy – Main Memory – Auxiliary Memory – Associative Memory – Cache Memory – Virtual Memory.

Input/output Organization: Input-Output Interface – Asynchronous Data Transfer – Modes of Transfer – Priority Interrupt – Direct Memory Access (DMA).

UNIT V

Pipeline and Multiprocessors

Pipeline: Parallel Processing – Pipelining – Arithmetic Pipeline – Instruction Pipeline.

Multiprocessors: Characteristics of Multiprocessors – Interconnection Structures – Inter Processor Arbitration – Inter Processor Communication and Synchronization.

Text Books:

1. *"Computer Systems Organization and Architecture", John D. Carpinelli, PEA, 2009.*
2. *"Computer Systems Architecture", 3/e, M. Moris Mano, PEA, 2007.*

Reference Books:

1. *"Computer Organization", Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5/e, MCG, 2002.*
2. *"Computer Organization and Architecture", 8/e, William Stallings, PEA, 2010.*
3. *"Computer Systems Architecture a Networking Approach", 2/e, Rob Williams.*
4. *"Computer Organization and Architecture" Ghoshal, Pearson Education, 2011.*
5. *"Computer Organization and Architecture", V. Rajaraman, T. Radakrishnan.*
6. *"Computer Organization and Design", P. Pal Chaudhuri, PHI*
7. *"Structured Computer Organization", Andrew S. Janenbaum, Todd Austin*
8. *"Computer Architecture" Parahmi, Oxford University Press*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem. (CSE)	T	Tu	C
(13A05402) DATABASE MANAGEMENT SYSTEMS		1	3

Course Objective:

- *To provide the student with clear conceptual understandings related to databases. After this course, the student should gain knowledge in the relational model, SQL, database design, storage & indexing, failure recovery and concurrency control.*

Learning Outcome:

- *Students can design the simple database, and can use the SQL instructions in developing the database applications.*
- *Can apply the ER concepts to design the databases.*
- *Advanced concepts like triggers, assertions and constraints can be applied effectively in designing the business applications.*

UNIT I

The Worlds of Database Systems -The Evolution of Database Systems - Overview of a Database Management System - Outline of Database System Studies.

The Entity-Relationship Model – Elements of E/R Model – Design Principles – The Modeling of Constraints – Weak Entity Sets.

The Relational Data Model – Basics of the Relational Model – From E/R Diagrams to Relational Designs – Converting Subclass Structures to Relations.

UNIT II

Functional Dependencies – Rules about Functional Dependencies -- Design of Relational Database Schemas – Multivalued Dependencies.

Relational Algebra and Calculus – Preliminaries, Relational algebra: Selection and Projection , Set Operations, Renaming, Joins, Division - Relational Calculus – Expressive power of Algebra and Calculus.

UNIT III

The Database Language SQL – Simple Queries in SQL – Queries Involving More than One Relation – Subqueries – Full Relation Operations – Database Modifications – Defining a Relation Schema in SQL – View Definitions - Transactions in SQL: Serializability, Atomicity, Transactions, Readonly Transactions, Dirty Reads, Other isolation levels.

Constraints and Triggers – Keys and Foreign keys – Constraints on Attributes and Tuples, Schema-level Constraints and Triggers.

UNIT IV

Representing Data Elements – Data Elements and Fields – Records – Representing Block and Record Addresses – Variable Length Data and Records – Record Modifications.

Index Structures – Indexes on Sequential Files – Secondary Indexes – B-Trees – Hash Tables.

UNIT V

Coping with System Failures – Issues and Models for Resilient Operation – Undo Logging – Redo Logging – Undo/Redo Logging – Protecting Against Media Failures.

Concurrency Control – Serial and Serializable Schedules – Conflict Serializability – Enforcing Serializability by Locks – Locking Systems with Several Lock Modes - Concurrency Control by Timestamps – Concurrency Control by Validation.

Text Books:

1. *"Database Systems, The Complete Book", Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom, 6th impression, 2011, Pearson.*
2. *"Data base Management Systems", Raghu Rama Krishnan, Johannes Gehrke, 3rd Edition, 2003, McGraw Hill.*

Reference Books:

1. *"Fundamentals of Database Systems", Elmasri Navrate, 6th edition, 2013, Pearson.*
2. *"Data base Systems design", Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.*
3. *"Introduction to Database Systems", C.J.Date, Pearson Education.*
4. *"Data base System Concepts", Silberschatz, Korth, McGraw Hill, V edition.*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II – II Sem. (CSE)	T	Tu	C
	3	1	3

(13A05403) JAVA PROGRAMMING

Course Objective:

- *Study the syntax, semantics and features of Java Programming Language*
- *Learn the method of creating Multi-threaded programs and handle exceptions*
- *Learn Java features to create GUI applications & perform event handling*

Learning Outcome:

- *Ability to solve problems using object oriented approach and implement them using Java*
- *Ability to write Efficient programs with multitasking ability and handle exceptions*
- *Create user friendly interface*

UNIT I

Introduction to Java: The key attributes of object oriented programming, Simple program, The Java keywords, Identifiers, Data types and operators, Program control statements, Arrays, Strings, String Handling

UNIT II

Classes: Classes, Objects, Methods, Parameters, Constructors, Garbage Collection, Access modifiers, Pass Objects and arguments, Method and Constructor Overloading, Understanding static, Nested and inner classes.

Inheritance – Basics, Member Access, Usage of Super, Multi level hierarchy, Method overriding, Abstract class, Final keyword.

Interfaces – Creating, Implementing, Using, Extending, and Nesting of interfaces.

Packages – Defining, Finding, Member Access, Importing.

UNIT III

Exception handling: Hierarchy, Fundamentals, Multiple catch clauses, Subclass exceptions, Nesting try blocks, Throwing an exception, Using Finally and Throws, Built-in exceptions, User-defined exceptions.

I/O: Byte streams and Classes, Character streams and Classes, Predefined streams, Using byte streams, Reading and Writing files using byte streams, Reading and writing binary data, Random-access files, File I/O using character streams, Wrappers.

UNIT IV

Multithreading: Fundamentals, Thread class, Runnable interface, Creating multiple threads, Life cycle of thread, Thread priorities, Synchronization, Thread communication, Suspending, Resuming and Stopping threads.

Applets: Basics, skeleton, Initialization and termination, Repainting, Status window, Passing parameters.

Networking: Basics, Networking classes and interfaces, InetAddress, Inet4Address and Inet6Address, TCP/IP Client Sockets, URL, URLConnection, HttpURLConnection, The URI class, Cookies, TCP/IP Server sockets, Datagrams.

UNIT V

Swings: The origin and design philosophy of swing, Components and containers, Layout managers, Event handling, Using a push button, jtextfield, jlabel and image icon, The swing buttons, Trees, An overview of jmenubar, jmenu and jmenuitem, Creating a main menu, Add mnemonics and accelerators to Menu items, showmessagedialog, showconfirmdialog, showinputdialog, showoptiondialog, jdialog, Create a modeless dialog.

Text Books:

1. *"Java Fundamentals - A Comprehensive Introduction", Herbert Schildt and Dale Skrien, Special Indian Edition, McGrawHill, 2013.*
2. *"Java The Complete Reference" Herbert Schildt, 8th Edition, 2011, Oracle press, TataMcGraw-Hill*

Reference Books:

1. *"Programming with Java" T.V.Suresh Kumar, B.Eswara Reddy, P.Raghavan Pearson Edition.*
2. *"Java – How to Program", Paul Deitel, Harvey Deitel, PHI.*
3. *"Core Java", Nageswar Rao, Wiley Publishers.*
3. *"Thinking in Java", Bruce Eckel, Pearson Education.*
4. *"A Programmers Guide to Java SCJP", Third Edition, Mughal, Rasmussen, Pearson.*
5. *"Head First Java", Kathy Sierra, Bert Bates, 0"Reilly*
6. *"SCJP – Sun Certified Programmer for Java Study guide" – Kathy Sierra, Bert Bates, McGrawHill*
7. *"Java in Nutshell", David Flanagan, 0"Reilly*
8. *"Core Java : Volume I – Fundamentals, Cay S. Horstmann, Gary Cornell, The Sun Micro Systems Press*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem (CSE)	T	Tu	C
	3	1	3

(13A05404) FORMAL LANGUAGES AND AUTOMATA THEORY

Course Objective:

- *Understand formal definitions of machine models.*
- *Classify machines by their power to recognize languages.*
- *Understanding of formal grammars, analysis*
- *Understanding of hierarchical organization of problems depending on their complexity*
- *Understanding of the logical limits to computational capacity*
- *Understanding of undecidable problems*

Learning Outcome:

At the end of the course, students will be able to

- *Construct finite state diagrams while solving problems of computer science*
- *Find solutions to the problems using Turing machines*
- *Design of new grammar and language*

UNIT I

Introduction: Basics of set theory, Relations on sets, Deductive proofs, Reduction to definitions, Other theorem forms, Proving equivalences about sets, The Contrapositive, Proof by contradiction, Counter examples, Inductive proofs, Alphabets, Strings, Languages, Problems, Grammar formalism, Chomsky Hierarchy

Finite Automata: An Informal picture of Finite Automata, Deterministic Finite Automata (DFA), Non Deterministic Finite Automata (NFA), Applying FA for Text search, Finite Automata with Epsilon transitions (ϵ-NFA or NFA- ϵ), Finite Automata with output, Conversion of one machine to another, Minimization of Finite Automata, Myhill-Nerode Theorem.

UNIT II

Regular Languages: Regular Expressions (RE), Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic laws for Regular Expressions, The Arden's Theorem, Using Arden's theorem to construct RE from FA, Pumping Lemma for RLs, Applications of Pumping Lemma, Equivalence of Two FAs, Equivalence of Two REs, Construction of Regular Grammar from RE, Constructing FA from Regular Grammar, Closure properties of RLs, Decision problem's of RLS, Applications of REs and FAs

UNIT III

Context Free Grammars and Languages: Definition of Context Free Grammars (CFG), Derivations and Parse trees, Ambiguity in CFGs, Removing ambiguity, Left recursion and Left factoring, Simplification of CFGs, Normal Forms, Linear grammars, Closure properties for CFLs, Pumping Lemma for CFLs, Decision problems for CFLs, CFG and Regular Language.

UNIT IV

Push Down Automata (PDA): Informal introduction, The Formal Definition, Graphical notation, Instantaneous description, The Languages of a PDA, Equivalence of PDAs and CFGs, Deterministic Push Down Automata, Two Stack PDA.

UNIT V

Turing Machines and Undecidability: Basics of Turing Machine (TM), Transitional Representation of TMs, Instantaneous description, Non Deterministic TM, Conversion of Regular Expression to TM, Two stack PDA and TM, Variations of the TM, TM as an integer function, Universal TM, Linear Bounded Automata, TM Languages, Unrestricted grammar , Properties of Recursive and Recursively enumerable languages, Undecidability, Reducibility, Undeciadable problems about TMs, Post's Correspondence Problem(PCP), Modified PCP.

Text Books:

1. *Introduction to Automata Theory, Formal Languages and Computation, Shyamalendu kandar, Pearson.*
2. *Introduction to Automata Theory, Languages, and Computation, Third Edition, John E.Hopcroft, Rajeev Motwani, Jeffery D. Ullman, Pearson.*

Reference Books:

1. *Introduction to Languages and the Theory of Computation, John C Martin, TMH, Third Edition.*
2. *Theory of Computation, Vivek Kulkarni, OXFORD.*
3. *Introduction to the Theory of Computation., Michel Sipser, 2nd Edition, Cengage Learning*
4. *Theory of computer Science Automata, Languages and Computation, K.L.P. Mishra, N. Chandrasekaran, PHI, Third Edition.*
5. *Fundamentals of the Theory of Computation, Principles and Practice, Raymond Greenlaw, H. James Hoover, Elsevier, Morgan Kaufmann.*
6. *Finite Automata and Formal Language A Simple Approach, A.M. Padma Reddy, Pearson*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem. (CSE)	T	Tu	C
(13A05405) PRINCIPLES OF PROGRAMMING LANGUAGES	3	1	3

Course Objective:

- *To study various programming paradigms.*
- *To provide conceptual understanding of High level language design and implementation.*
- *To introduce the power of scripting languages*

Learning Outcome:

- *Select appropriate programming language for problem solving*
- *Design new programming language.*
- *Gain Knowledge and comparison of the features of programming languages*

UNIT I

Preliminary Concepts: Reasons for studying, Programming domains, Language Evaluation Criteria, Influences on Language design, Language categories, Language design Trade-offs, Implementation methods, Programming environments.

Syntax and Semantics: Introduction, General problem of describing syntax, Formal methods of describing syntax, Describing the meaning of programs – Dynamic semantics.

Introduction to Programming concepts: Names, Variables, The concept of binding, Type checking, Strong typing, Type compatibility, Scope, Scope and lifetime, Referencing environments, Named constants

UNIT II

Data types: Introduction, primitive, Character string, user defined ordinal, array, associative array, record, union, pointer and reference types

Expressions: Arithmetic relational and Boolean expressions, Type conversions, Short circuit evaluation, Assignment Statements, Mixed-mode arithmetic.

Control Structures – Selection, Iterative, Unconditional branching, guarded commands.

UNIT III

Subprograms: Fundamentals of sub-programs, Design issues of subprograms, Local referencing environments, Parameter passing methods, Generic sub-programs: Generic functions in C++, Generic methods in Java, Design issues for functions, Coroutines, General semantics of Calls and Returns, Implementing Simple subprograms, Implementing subprograms with Stack-Dynamic Local variables, Nested subprograms.

UNIT IV

Concurrency: Why concurrency, Programs and processes, Problems with concurrency, Process interactions, Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads, statement level concurrency.

Exception handling: Exceptions, exception Propagation, Exception handling in Java.

Logic Programming: Introduction, Introduction to Predicate calculus, Predicate calculus and proving theorems, Overview of logic programming, Origins of prolog, Basic elements of prolog, Deficiencies of prolog, Applications of logic programming

UNIT V

Functional Programming Languages: Introduction, Mathematical functions, Fundaments of functional programming languages, Fundamentals of LISP, Common lisp, Applications of Functional languages, Comparison of Functional and imperative languages.

Scripting Language: Pragmatics, Key Concepts, Case Study : Python – Values and Types, Variables , Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library

Text Books:

1. *Concepts of Programming Languages, Robert .W. Sebesta 10/e, Pearson Education,2008.*
2. *Programming Language Design Concepts, D. A. Watt, Wiley dreamtech,rp-2007.*

Reference Books :

1. *Programming Languages, 2nd Edition, A.B. Tucker, R.E. Noonan, TMH.*
2. *Programming Languages, K. C.Louden, 2nd Edition, Thomson, 2003.*
3. *LISP, Patric Henry Winston and Paul Horn, Pearson Education.*
4. *Programming in Prolog, W.F. Clocksin,& C.S.Mellish, 5th Edition, Springer.*
5. *Programming Python, M.Lutz, 3rd Edition, O"reilly, SPD, rp-2007.*
6. *Core Python Programming, Chun, II Edition, Pearson Education, 2007.*
7. *Guide to Programming with Python, Michael Dawson, Thomson, 2008*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem (CSE)	T	Tu	C
	3	1	3

(13A05406) DESIGN AND ANALYSIS OF ALGORITHMS

Course Objective:

- *To know the importance of the complexity of a given algorithm.*
- *To study various algorithmic design techniques.*
- *To utilize data structures and/or algorithmic design techniques in solving new problems.*
- *To know and understand basic computability concepts and the complexity classes P, NP, and NP-Complete.*
- *To study some techniques for solving hard problems.*

Learning Outcome:

- *Analyze the complexity of the algorithms*
- *Use techniques divide and conquer, greedy, dynamic programming, backtracking, branch and bound to solve the problems.*
- *Identify and analyze criteria and specifications appropriate to new problems, and choose the appropriate algorithmic design technique for their solution.*
- *Able to prove that a certain problem is NP-Complete.*

UNIT I

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

Divide and Conquer: General method, Binary Search, Finding the maximum and minimum, Merge sort, Quick Sort, Selection sort, Stressen's matrix multiplication.

UNIT II

Greedy Method: General method, Knapsack problem, Job Scheduling with Deadlines, Minimum cost Spanning Trees, Optimal storage on tapes, Single-source shortest paths.

Dynamic programming: General Method, Multistage graphs, All-pairs shortest paths, Optimal binary search trees, 0/1 knapsack, The traveling sales person problem.

UNIT III

Basic Traversal and Search Techniques: Techniques for binary trees, Techniques for Graphs, Connected components and Spanning trees, Bi-connected components and DFS

Back tracking: General Method, 8 – queens problem, Sum of subsets problem, Graph coloring and Hamiltonian cycles, Knapsack Problem.

UNIT IV

Branch and Bound: The method, Travelling salesperson, 0/1 Knapsack problem, Efficiency considerations.

Lower Bound Theory: Comparison trees, Lower bounds through reductions – Multiplying triangular matrices, Inverting a lower triangular matrix, Computing the transitive closure.

UNIT V

NP – Hard and NP – Complete Problems: NP Hardness, NP Completeness, Consequences of being in P, Cook's Theorem, Reduction Source Problems, Reductions: Reductions for some known problems

Text Books:

1. *"Fundamentals of Computer Algorithms", Ellis Horowitz, S. Satraj Sahani and Rajasekhran, 2nd edition, 2012, University Press.*
2. *"Design and Analysis of Algorithms", Parag Himanshu Dave, Himanshu Bhalchandra Dave, Second Edition, 2009, Pearson Education.*

Reference Books :

1. *"Introduction to Algorithms", secondedition,T.H.Cormen,C.E.Leiserson, R.L.Rivest,and C.Stein,PHI Pvt. Ltd./ Pearson Education*
2. *"Introduction to Design and Analysis of Algorithms A strategic approach", R.C.T.Lee, S.S.Tseng, R.C.Chang and T.Tsai, Mc Graw Hill.*
3. *"Data structures and Algorithm Analysis in C++", Allen Weiss, Second edition, Pearson education.*
4. *"Design and Analysis of algorithms", Aho, Ullman and Hopcroft,Pearson education.*
5. *"Algorithms" – Richard Johnson baugh and Marcus Schaefer, Pearson Education*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem. (CSE) L 3 C 2

(13A05407) DATABASE MANAGEMENT SYSTEMS LAB

Course Objective:

- *To create a database and query it using SQL, design forms and generate reports.*
- *Understand the significance of integrity constraints, referential integrity constraints, triggers, assertions.*

Learning Outcome:

- *Design databases*
- *Retrieve information from data bases*
- *Use procedures to program the data access and manipulation*
- *Create user interfaces and generate reports*

LIST OF EXPERIMENTS:

1. Practice session: Students should be allowed to choose appropriate DBMS software, install it, configure it and start working on it. Create sample tables, execute some queries, use SQLPLUS features, use PL/SQL features like cursors on sample database. Students should be permitted to practice appropriate User interface creation tool and Report generation tool.

2. A college consists of number of employees working in different departments. In this context, create two tables **employee** and **department**. Employee consists of columns empno, empname, basic, hra, da, deductions, gross, net, date-of-birth. The calculation of hra,da are as per the rules of the college. Initially only empno, empname, basic have valid values. Other values are to be computed and updated later. Department contains deptno, deptname, and description columns. Deptno is the primary key in department table and referential integrity constraint exists between employee and department tables. Perform the following operations on the the database:

    - Create tables department and employee with required constraints.
    - Initially only the few columns (essential) are to be added. Add the remaining columns separately by using appropriate SQL command
    - Basic column should not be null
    - Add constraint that basic should not be less than 5000.
    - Calculate hra,da,gross and net by using PL/SQL program.
    - Whenever salary is updated and its value becomes less than 5000 a trigger has to be raised preventing the operation.
    - The assertions are: hra should not be less than 10% of basic and da should not be less than 50% of basic.
    - The percentage of hra and da are to be stored separately.

- When the da becomes more than 100%, a message has to be generated and with user permission da has to be merged with basic.
- Empno should be unique and has to be generated automatically.
- If the employee is going to retire in a particular month, automatically a message has to be generated.
- The default value for date-of-birth is 1 jan, 1970.
- When the employees called daily-wagers are to be added the constraint that salary should be greater than or equal to 5000 should be dropped.
- Display the information of the employees and departments with description of the fields.
- Display the average salary of all the departments.
- Display the average salary department wise.
- Display the maximum salary of each department and also all departments put together.
- Commit the changes whenever required and rollback if necessary.
- Use substitution variables to insert values repeatedly.
- Assume some of the employees have given wrong information about date-of-birth. Update the corresponding tables to change the value.
- Find the employees whose salary is between 5000 and 10000 but not exactly 7500.
- Find the employees whose name contains 'en'.
- Try to delete a particular deptno. What happens if there are employees in it and if there are no employees.
- Create alias for columns and use them in queries.
- List the employees according to ascending order of salary.
- List the employees according to ascending order of salary in each department.
- Use '&&' wherever necessary
- Amount 6000 has to be deducted as CM relief fund in a particular month which has to be accepted as input from the user. Whenever the salary becomes negative it has to be maintained as 1000 and the deduction amount for those employees is reduced appropriately.
- The retirement age is 60 years. Display the retirement day of all the employees.
- If salary of all the employees is increased by 10% every year, what is the salary of all the employees at retirement time.
- Find the employees who are born in leap year.
- Find the employees who are born on feb 29.
- Find the departments where the salary of atleast one employee is more than 20000.
- Find the departments where the salary of all the employees is less than 20000.
- On first January of every year a bonus of 10% has to be given to all the employees. The amount has to be deducted equally in the next 5 months. Write procedures for it.
- As a designer identify the views that may have to be supported and create views.

- As a designer identify the PL/SQL procedures necessary and create them using cursors.
- Use appropriate Visual programming tools like oracle forms and reports, visual basic etc to create user interface screens and generate reports.

Note: As a designer identify other operations that may be required and add to the above list. The above operations are not in order. Order them appropriately. Use SQL or PL/SQL depending on the requirement.

3. Students may be divided into batches and the following experiments may be given to them to better understand the DBMS concepts. Students should gather the required information, draw ER diagrams, map them to tables, normalize, create tables, triggers, procedures, execute queries, create user interfaces, and generate reports.

    - Student information system
    - APSRTC reservation system
    - Hostel management
    - Library management
    - Indian Railways reservation
    - Super market management
    - Postal system
    - Banking system
    - Courier system
    - Publishing house system

References:

1. *"Learning Oracle SQL and PL/SQL", Rajeeb C. Chatterjee, PHI.*
2. *"Oracle Database 11g PL/SQL Programming", M.Mc Laughlin,TMH.*
3. *"Introduction to SQL", Rick F.Vander Lans, Pearson education.*
4. *"Oracle PL/SQL", B.Rosenzweig and E.Silvestrova, Pearson education.*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem. (CSE) L C
3 2

(13A05408) JAVA PROGRAMMING LAB

Week-1:

1. Use Eclipse or Netbean platform and acquaint with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with java program to find prime numbers between 1 to n.

Week-2:

1. Write a Java program that prints all real and imaginary solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula.
2. Write a Java program for sorting a given list of names in ascending order
3. Write a java program to accept a string from user and display number of vowels, consonants, digits and special characters present in each of the words of the given text.

Week -3:

1. Write a java program to make rolling a pair of dice 10,000 times and counts the number of times doubles of are rolled for each different pair of doubles.
   Hint: Math.random()
2. Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read display the complete set of unique values input after the user enters each new value.
3. Write a java program to read the time intervals (HH:MM) and to compare system time if the system time between your time intervals print correct time and exit else try again to repute the same thing. By using StringToknizer class

Week-4:

1. Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part<n> where n is the sequence number of the part file.
2. Write java program to create a super class called Figure that receives the dimensions of two dimensional objects. It also defines a method called area that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and second is Triangle. Each of the sub class overridden area() so that it returns the area of a rectangle and a triangle respectively.
3. Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds

Week-5:

1. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication
2. Write a java program to find and replace pattern in given file,
3. Use inheritance to create an exception super class called EexceptionA and exception sub class ExceptionB and ExceptionC, where ExceptionB inherits from ExceptionA and

ExceptionC inherits from ExceptionB. Write a java program to demonstrate that the catch block for type ExceptionA catches exception of type ExceptionB and ExceptionC

Week-6:

1. Write a java program to convert an ArrayList to an Array.
2. Write a Java Program for waving a Flag using Applets and Threads
3. Write a Java Program for Bouncing Ball (The ball while moving down has to increase the size and decrease the size while moving up)

Week-7:

1. Write a Java Program for stack operation using Buttons and JOptionPane input and Message dialog box.
2. Write a Java Program to Addition, Division, Multiplication and subtraction using JOptionPane dialog Box and Textfields.

Week-8:

1. Write a Java Program for the blinking eyes and mouth should open while blinking.
2. Implement a Java Program to add a new ball each time the user clicks the mouse. Provided a maximum of 20 balls randomly choose a color for each ball.

Week-9:

1. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Jtable component
2. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.

Week-10:

1. Write a Java Program to implement the opening of a door while opening man should present before hut and closing man should disappear.
2. Write a Java code by using JtextField to read decimal value and converting a decimal number into binary number then print the binary value in another JtextField

Week-11:

1. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result.
2. Write a Java program for handling mouse events.

Week-12:

1. Write a java program establish a JDBC connection, create a table student with properties name, register number, mark1,mark2, mark3. Insert the values into the table by using the java and display the information of the students at front end.

Note: In addition to the above experiments, the instructor may identify the experiments in the important concepts like Multi Threading (Producer Consumer Problem etc.) and Networking (Client-Server problem etc.).

Text Books:

1. *Java How to Program, Sixth Edition, H.M.Dietel and P.J.Dietel, Pearson Education/PHI*
2. *Java The Complete Reference" by Herbert Schildt, TMH, 8th Edition*

Reference Books:

1. *Introduction to Java programming, Sixth edition, Y.Daniel Liang, Pearson Education*
2. *Programming in Java, Sachine*
3. *Big Java, 2nd edition, Cay Horstmann, Wiley Student Edition, Wiley India Private Limited.*
4. *Introduction to Programming with Java, J.Dean & R.Dean, McGraw Hill education.*
5. *Java Programming, D S Malik, Cengage Learning, India Edition*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. II - II Sem. (CSE) T
2

(13A52301) HUMAN VALUES & PROFESSIONAL ETHICS (AUDIT COURSE)

Course Objective:

- *This course deals with professional ethics which includes moral issues and virtues, social responsibilities of an engineer, right, qualities of Moral Leadership*

UNIT I
ENGINEERING ETHICS
Senses of 'Engineering Ethics' – Variety of Moral Issues – Types of Inquiry – Moral Dilemmas – Moral Autonomy – Kohlberg's Theory – Gilligan's Theory – Consensus and Controversy – Professions and Professionalism – Professional Ideals and Virtues – Uses of Ethical Theories

UNIT II
ENGINEERING AS SOCIAL EXPERIMENTATION
Engineering as Experimentation – Engineers as Responsible Experimenters – Research Ethics – Codes of Ethics – Industrial Standards – A Balanced Outlook on Law – The Challenger Case Study

UNIT III
ENGINEER'S RESPONSIBILITY FOR SAFETY
Safety and Risk – Assessment of Safety and Risk – Risk Benefit Analysis – Reducing Risk – The Government Regulator's Approach to Risk – Chernobyl Case Studies and Bhopal

UNIT IV
RESPONSIBILITIES AND RIGHTS
Collegiality and Loyalty – Respect for Authority – Collective Bargaining – Confidentiality– Conflicts of Interest – Occupational Crime – Professional Rights – Employee Rights –Intellectual Property Rights (IPR) – Discrimination

UNIT V
GLOBAL ISSUES
Multinational Corporations – Business Ethics – Environmental Ethics – Computer Ethics - Role in Technological Development – Weapons Development – Engineers as Managers – Consulting Engineers – Engineers as Expert Witnesses and Advisors – Honesty –Moral Leadership – Sample Code of Conduct

Text Books:

1. *Mike Martin and Roland Schinzinger, "Ethics in Engineering", McGraw Hill, New York 2005.*
2. *Charles E Harris, Michael S Pritchard and Michael J Rabins, "Engineering Ethics – Concepts and Cases", Thompson Learning, 2000.*

Reference Books:

1. *Charles D Fleddermann, "Engineering Ethics", Prentice Hall, New Mexico, 1999.*
2. *John R Boatright, "Ethics and the Conduct of Business", Pearson Education, 2003.*
3. *Edmund G Seebauer and Robert L Barry, "Fundamentals of Ethics for Scientists and Engineers", Oxford University Press, 2001.*
4. *Prof. (Col) P S Bajaj and Dr. Raj Agrawal, "Business Ethics – An Indian Perspective", Biztantra, New Delhi, 2004.*
5. *David Ermann and Michele S Shauf, "Computers, Ethics and Society", Oxford University Press, 2003.*