# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING

## COURSE STRUCTURE & SYLLABUS for R-17 III YEAR

### III YEAR I SEMESTER

| S.No | Course code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|---|
| 1 | A55PC1 | Computer Networks | 4 | 1 | 0 | 4 |
| 2 | A55PC2 | Principles of Programming Languages | 4 | 1 | 0 | 4 |
| 3 | A55PC3 | Design and Analysis of Algorithms | 4 | 1 | 0 | 4 |
| 4 | A55PC4 | Compiler Design | 3 | 1 | 0 | 3 |
| 5 | A55PE5 | 1. Software Engineering<br>2. Computer Graphics<br>3. Image Processing and Pattern Recognition | 3 | 0 | 0 | 3 |
| 6 | | (Open Elective -1) | 3 | 1 | 0 | 3 |
| 7 | A55PC6 | Computer Networking Lab | 0 | 0 | 3 | 2 |
| 8 | A55PC7 | Compiler Design Lab | 0 | 0 | 3 | 2 |
| 9 | A55HS8 | Advanced Communication Skills Lab | 0 | 0 | 3 | 2 |
| | | **Total Credits** | **21** | **5** | **9** | **27** |

### III YEAR II SEMESTER

| S.No | Course code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|---|
| 1 | A56PC1 | Web Technologies | 4 | 1 | 0 | 4 |
| 2 | A56PC2 | Data Warehousing and Data Mining | 4 | 1 | 0 | 4 |
| 3 | A56PC3 | Object Oriented Analysis and Design | 4 | 0 | 0 | 4 |
| 4 | A56PE4 | 1. Introduction to Analytics.<br>2. Information Security & Management<br>3. Cloud Computing | 3 | 0 | 0 | 3 |
| 5 | A56PE5 | 1. Software Testing Methodologies<br>2. Artificial Intelligence<br>3. Semantic Web and Social Networks | 3 | 0 | 0 | 3 |
| 6 | | Open Elective -2 | 3 | 0 | 0 | 3 |
| 7 | A56PC7 | Technical Skills Enhancement Lab | 0 | 1 | 3 | 2 |
| 8 | A56PC8 | Case Tools and Web Technologies Lab | 0 | 0 | 3 | 2 |
| 9 | A56PC9 | Data Warehousing and Data Mining Lab | 0 | 0 | 3 | 2 |
| | | **Total Credits** | **21** | **3** | **9** | **27** |

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**B.TECH. COMPUTER SCIENCE AND ENGINEERING – R17**

## COMPUTER NETWORKS (A55PC1)

**B.Tech III Year I Semester** **L/T/P/C**
**4/1/0/4**

**COURSE OBJECTIVE**:
Explore the basics of computer networks various layers and their protocols.

**COURSE OUTCOMES**: After completion of the course the student will be able to
1. Learn the fundamentals of computer networks - L2
2. Construct error free DLL –L3
3. Compare and construct various routing protocols-L3
4. Compare the TCP and UDP protocols-L3
5. Demonstrate different application layer protocol-L4

**UNIT I**

**INTRODUCTION TO COMPUTER NETWORKS**

Components, Direction of Data flow Types of connections, topologies, protocols and standards of ISO/OSI model, TCP/IP Model.

**PHYSICAL LAYER**

Transmission modes, Multiplexing, Transmission media, Switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

**UNIT II**

**DATA LINK LAYER**

Introduction, Framing, Error Detection and Correction-Parity-LRC-CRC Hamming code, flow and error control, Noiseless channels, Noisy Channels, HDLC, Point to Point Protocols.
Medium Access Sub Layer: ALOHA, CSMA/CD, LAN-Ethernet IEEE802.5, IEEE 802.11, Random Access, Controlled Access, Channelization.

**UNIT III NETWORK LAYER**

Logical Addressing, Internetworking, Tunneling, Address mapping, ICMP, IGMP, Forwarding, Unit-Cast Routing Protocols, Multicast Routing Protocols.

**UNIT IV TRANSPORT LAYER**

Process to Process Delivery, UDP and TCP protocols, Data traffic, congestion, congestion control, QoS in switched networks.

**UNIT V**

**APPLICATION LAYER:**

Domain name space, DNS in Internet, Electronic Mail, SMPT, FTP, WWW, HTTP, SNMP

**TEXT BOOKS:**


1. Data Communications and Networking, Behrouz. A. Forouzan, Fourth Edition TMH, 2006.
2. Computer Networks, Andrew S Tanenbaum 4th Edition Pearson Education, PHI


**REFERENCES:**


1. Data Communication and Computer Networks, P.C. Gupta, PHI
2. An Engineering approach to Computer Networks, S. Keshav, 2nd Edition Pearson Education.
3. Understanding communications and Networks, 3rd edition, W.A. Shay, Cengage Learning.
4. Computer Networking: A. Top-Down Approach Featuring the Internet. James F Kurose & Keith W. Ross. 3rd Edition Pearson Education.
5. Data and Computer Communication, William Stallings, Sixth Edition, Pearson Education, 2000.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING –R17

## PRINCIPLES OF PROGRAMMING LANGUAGES (A55PC2)

**B.Tech III Year I Semester** **L/T/P/C 4/1/0/4**

**COURSE OBJECTIVE:**

Gain insights of programming constructs and implementation issues at high level language design.

**COURSE OUTCOMES:**

After completion of the course the student will be able to

1. Understand various programming paradigms.- L2
2. Discussion of various language data types and statements - L2
3. Analyze various languages sub program constructs - L3
4. Contrast and compare concurrency control mechanism in various languages – L4
5. Demonstration of functional and scripting language – L3

**UNIT - I :**

**Preliminary Concepts:** Concepts of Programming Languages, Programming domains, Language Evaluation Criteria, Influence of Language Design, Language Categories, Programming Paradigms-imperative, Object Oriented, functional Programming, Implementation methods, Programming Environments.

**Syntax and Semantics:** General problem of describing syntax and semantics, formal methods of describing syntax-BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.

**UNIT – II :**

**Data Types:** Introduction, primitive, character, user defined, array, associative, record, union, pointer, and reference types. Design and implementation uses related to these types. Names, variable, concept binding, type checking, strong typing, type compatibility, name constants, variable initialization.

**Expression and Statements:** Arithmetic relational and Boolean Expressions, short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures- Statement level, Compound Statements, Selection, Iteration, Unconditional Statements, Guarded commands.

**UNIT - III :**

**Subprograms and Blocks :** Fundamentals of sub-programs, scope of life time of variables, static and dynamic scope design issues of subprograms and operations. Local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co-routines.

**UNIT - IV :**

**Abstract Data Types :** Abstraction and Encapsulation, Introduction to data abstraction, design issues, language examples, c++ parameterized ADT, C++, and Java.

**Concurrency :** Sub program level concurrency, semaphores, monitors, message passing, and java threads.

**Exception Handling :** Exceptions, exception propagation, Exception handler in C++, Java

**UNIT – V :**

**Functional Programming Languages :** Introduction, fundamentals, of FPL, LISP, ML, Haskell, application of functional programming languages. Comparison of functional and imperative languages.

**Python Scripting :** Introduction to Python, Using the Python Interpreter, More Control Flow Tools, Data Structures, Modules, Input and Output, Errors and Exceptions, Classes, Brief Tour of the Standard Library.

**TEXT BOOKS:**


1. Concepts of Programming Languages Robert. W. Sebesta 8/e, Pearson Education 2008.
2. Programming Language Design Concepts, D.A. Watt, Wiley dreamtech rp-2007.
3. Python Tutorial by Guido Van Rossum, Fred L. Drake Jr. editor release 2.6.4.


**REFERENCE BOOKS:**


1. Programming Languages, 2nd edition, A.B.Tucker, R.E. Noonan, TMH.
2. LISP Patric Henry Winston and Paul Horn Pearson Education.
3. Programming in Prolog, W.F. Clocksin & C.S. Mellish, 5th Edition Springer.
4. Core Python Programming chun 2nd edition, Pearson Education,2007.
5. Guide to Programming with Python, Michel Dawson, Thomson, 2008.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**B.TECH. COMPUTER SCIENCE AND ENGINEERING – R17**

## DESIGN AND ANALYSIS OF ALGORITHMS (A55PC3)

**B.Tech III Year I Semester**

**Course Objective**:

Understand the design paradigms for developing an algorithm and analyzing it for a given problem.

**Course Outcomes**:

After learning the contents of this course, the student will be able to

1. Apply mathematical analysis methods to analyze the performance of algorithms and apply divide and conquer technique to solve the computing problems. L3
2. Demonstrate disjoint set operations and apply back tracking technique to solve the computing problems. L3
3. Apply Greedy method to solve various computing problems. – L3
4. Synthesize efficient algorithms in common engineering design situations using dynamic programming technique. L5
5. Solve complex problems using branch and bound technique and analyze NP hard and NP complete problems - L4

## UNIT I:

**INTRODUCTION**

Algorithm definition, Algorithm specification, Performance Analysis, Space Complexity, Time Complexity, Randomized Algorithms.

**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-Queens Problem, sum of subsets problem, graph coloring, Hamilton 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 PROBLEM**

Basic concepts, Non-deteministic algorithms, NP-Hard, and NP-Complete classes, Cook's Theorem.

## TEXT BOOKS:


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


## REFERENCES:


1. Algorithms Design: Foundations, Analysis and Internet examples, M.T. Goodrich and R. Tamassia, John Wiley and Sons.
2. Design and Analysis of Algorithms, S. Sridhar Oxford Univ. Press.
3. Introduction to Algorithms 3rd Edition, T.H. Cormen, C.E.Leiserson, R.L. Rivest and C. Stein, PHI.
4. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson Education.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**
**B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17**

## COMPILER DESIGN (A55PC4)

**B.Tech III Year I Semester** **L/T/P/C 3/1/0/3**

## COURSE OBJECTIVE:

Learn the process of designing and developing a compiler for a target machine.

## COURSE OUTCOMES:

After completion of the course the student will be able to

1. Understand different phases of a compiler – L2
2. Design different parsers – L4
3. Demonstrate syntax-directed translation schemes and generate intermediate code- L3
4. Analyze code optimization techniques , and runtime environment – L4
5. Design machine independent code optimization techniques – L5

## UNIT I :

**INTRODUCTION**

Language Processors, the structure of compiler, the science of building a compiler, Programming Language basics.

**LEXICAL ANALYSIS**

The role of Lexical Analyser, Input buffering, Recognition of Tokens, The Lexical-Analyzer Generator, Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyser 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, and Parser Generators.

## UNIT III:

**SYNTAX-DIRECTED TRANSLATIONS**

Syntax-Directed Definitions, Evaluation Orders for SDD's Applications of Syntax-Directed Translations, 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 Non-Local 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 largest 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 Jeffry D. Ullman, Pearson.

## REFERENCES:

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. Lex & Yacc John R Levine, Tony Mason Doug Brown O'reilly

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17**

## SOFTWARE ENGINEERING (A55PE5)

**B.Tech III Year I Semester** **L/T/P/C**
**3/0/0/3**

## COURSE OBJECTIVE:

To know about the software process models, SRS document, architectural styles, verification and validation of developed code and application prototype.

## COURSE OUTCOMES:

After completion of course the student will be able to

1. Understand minimum requirements for the development of application and different software process models –L2
2. Demonstrate various software requirement and system models –L3
3. Apply appropriate software architectures and conducting component level design – L3
4. Apply the process of validation and verification for a developed application(Prototype) – L3
5. Discover various software risks in any software applications –L3

## UNIT I:

**INTRODUCTION TO SOFTWARE ENGINEERING**

The evolving role of software, Changing Nature of Software, legacy software, software myths. A GENERIC VIEW OF PROCESS

Software engineering- A layered technology, a process framework, The capability Maturity Model Integration (CMMI), process patterns, process assessment, personal and team process models.

**PROCESS MODELS**

The Waterfall Model, Incremental Process Model, Evolutionary Process models, specialized process models, unified process.

## UNIT II:

**SOFTWARE REQUIREMENTS**

Functional and Non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

**REQUIREMENTS ENGINEERING PROCESS**

Feasibility studies, Requirements elicitation and analysis, requirements validation, requirements management.

**SYSTEM MODELS**

Context Models, Behavioral Models, Data Models, Object Models, structured methods.

## UNIT III:

**DESIGN ENGINEERING**

Design Process and Design quality, Design concepts, the design model, pattern based software design.

**CREATING AN ARCHITECTURAL DESIGN**

Software architecture, Data Design, Architectural styles and patterns. Architectural Design, assessing alternative architectural designs, mapping data flow into a software architecture.

**MODELING COMPONENT-LEVEL DESIGN**

Design class-based components, conducting component-level design, object constraint language, design conventional components.

## UNIT IV:

**TESTING STRATEGIES**

A strategic approach to software testing, testing strategies for conventional software, Baclk-Box and White-Box testing. Validation testing, system testing, the art of debugging.

**PRODUCT METRICS**

Software Quality, Frame work for Product metrics, Metric for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

**METRICS FOR PROCESS AND PRODUCTS**

Software Measurement, Metrics for Software Quality.

**UNIT V:**

## RISK MANAGEMENT

Reactive versus Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement. RMMM, RMMM plan.

## TEXT BOOKS:


1. Software engineering A Practitioner's approach, Roger S Pressman, Sixth Edition McGraw Hill International Edition.
2. Software Engineering: Ian Sommerville, Seventh Edition, Pearson Education.


## REFERENCES:


1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India 2010
2. Software Engineering: A Primer, Waman S. Jawadekar, Tata McGraw Hill, 2008
3. Software Engineering Foundations, Yingxu Wang, Auerbach Publications 2008

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17**

## COMPUTER GRAPHICS (A55PE5)

**B.Tech III Year I Semester**

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

## COURSE OBJECTIVE:

To learn and make understand about fundamentals of Graphics to construct design animated scenes for virtual object creations, present the content graphically.

## COURSE OUTCOMES:

1. Students can animate scenes entertainment
2. Will be able work in computer aided design for content presentation
3. Better analogy data with pictorial representation.

## UNIT I:

### INTRODUCTION

Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices output primitives: Points and lines, Line drawing algorithms, mid point circle and ellipse algorithms. Filled area primitives: scan line polygon fill algorithm, boundary-fill and flood fill algorithms.

## UNIT II:

### 2-D GEOMETRICAL TRANSFORMS

Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformation between coordinate systems.

2-D Viewing: The viewing pipeline, viewing coordinate reference frame, window to viewport coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus beck line clipping algorithms, Sutherland-Hodgeman Polygon clipping algorithm

## UNIT III:

### OBJECT REPRESENTATION

Polygon surfaces, quadric surfaces, spline representation, Hermite Curve, Bezier Curve and B-Spline curves, Bezier and B-Spline surfaces, sweep representations, octrees BSP trees, 3-D Geometric Transformations: Translation, Rotation, Scaling, Reflection and Shear transformations, composite transformations, 3-D viewing, viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

## UNIT IV:

### VISIBLE SURFACE DETECTION METHODS

Classification, back-face detection, depth-buffer, scaline, depth sorting, BSP- tree methods, are sub-division and octree methods Illumination models and surface rendering methods: Basic illumination models, polygon rendering methods.

## UNIT V:

### COMPUTER ANIMATION

Design of animation sequence general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

## TEXT BOOKS:


1. COMPUTER GRAPHICS C VERSION by Donald Hearn and M. Pauline Baker, Pearsosn Education.
2. Computer Graphics second edition "Zhigand Xiang, Roy Plastock, Schaum's outlines Tata Mc Graw hill edition.


## REFERENCE BOOKS:


1. Computer Graphics Principles & Practice", Second Edition in C, Foley, VanDam, Feiner and Hughes, Pearson Education.
2. Procedural Elements for Computer Graphics, David F Rogers, Tata McGraw Hill, 2nd edition.
3. Principles of Interactive Graphics, Neuman and Sproul, TMH
4. An Integrated Introduction to Computer Graphics and Geometric Modelling, R. Goldman, CRC Press, Taylor & Francis Group.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING – R17

## IMAGE PROCESSING AND PATTERN RECOGNITION (A55PE5)

**B.Tech III Year I Semester** **L/T/P/C**
**3/0/0/3**

**COURSE OBJECTIVE:**

To Learn the techniques used in processing images and Pattern Recognition.

**COURSE OUTCOMES:**

After completion of course the student will be able to

1. Learn the fundamental steps of Image Processing, and interpret the filtering approaches.
2. Demonstrate the operations of morphology, illustrate segmentation and Edge detection operations.
3. Apply and bulid image compression techniques.
4. Name the techniques involved in Representation and Description.
5. Explain the fundamental pattern recognition concepts and classify them.

**UNIT I:**

Fundamental steps of image processing, components of an image processing of system. The image model and image acquisition, sampling and quantization, relationship between pixels, distance functions, scanner.

Statistical and spatial operations, Intensity functions transformations, histogram processing, smoothing & sharpening — spatial filters Frequency domain filters, homomorphic filtering, image filtering & restoration, Inverse and weiner filtering, FIR weiner filter, Filtering using image transforms, smoothing splines and interpolation.

**UNIT II:**

Morphological and other area operations, basic morphological operations, opening and closing operations, dilation erosion, Hit or Miss transform, morphological algorithms, extension to grey scale images.

Segmentation and Edge detection region operations, basic edge detection, second order detection, crack edge detection, gradient operators, compass and Laplace operators, edge linking and boundary detection, thresholding, region based segmentation, segmentation by morphological watersheds.

**UNIT III:**

**Image compression**

Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression, image data compression-predictive technique, pixel coding, transfer coding theory, lossy and lossless predictive type coding, Digital Image Water marking.

**UNIT IV:**

**Representation and Description**

Chain codes, Polygonal approximation, Signature Boundary Segments, Skeltons, Boundary Descriptors, Regional Descriptors, Relational Descriptors, Principal components for Description, Relational Descriptors

**UNIT- V:**

**Pattern Recognition Fundamentals**

Basic Concepts of pattern recognition, Fundamental problems in pattern recognition system, design concepts and methodologies, example of automatic pattern recognition systems, a simple automatic pattern recognition model

**Pattern classification:** Pattern classification by distance function: Measures of similarity, Clustering criteria, K-means algorithm, Pattern classification by likelihood function: Pattern classification as a Statistical decision problem, Bayes classifier for normal patterns.

**TEXT BOOKS:**


1. Digital Image Processing Third edition, Pearson Education, Rafael C. Gonzalez, Richard E. Woods.
2. Pattern recognition Principles: Julus T. Tou, and Rafel C. Gonzalez, Addision-Wesly Publishing Company.
3. Digital Image Processing, M.Anji Reddy, Y.Hari Shankar, BS Publications.


**REFERENCE BOOKS:**


1. Image Processing, Analysis and Machine Vision, Second Edition, Milan Sonka, Vaclav Hlavac and Roger Boyle. Thomson learning
2. Digital Image Processing — William k. Pratl —John Wiley edition.
3. Fundamentals of digital image processing — by A.K. Jam, PHI.
4. Pattern classification, Richard Duda, Hart and David strok John Wileypublishers.
5. Digital Image Processing, S.Jayaraman,S. Esakkirajan, T.Veerakumar, TMH.
6. Pattern Recognition, R.Shinghal, Oxford University Press.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17

## COMPUTER NETWORKING LAB (A55PC6)

**B.Tech III Year I Semester** **L/T/P/C**
**0/0/3/2**

**COURSE OBJECTIVE:**

To understand the functionalities of various layers of OSI model and implement encryption and decryption algorithm.

**COURSE OUTCOMES:**

After completion of the course the student will be able to

1. Implement different DLL framing methods and CRC polynomials –L3
2. Apply appropriate algorithm for the finding of shortest route – L3
3. Configure the routing table – L3
4. Compare and contrast different encryption and decryption algorithms – L4

**System/ Software Requirement**

Intel based desktop PCs LAN connected with minimum of 166 MHZ or faster process with at least 64 MB RAM and 100 MB free disk space.

1. Implementing the data link layer framing methods such as 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 through a graph.
4. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table art each node using distance vector routingalgorithm.
5. Take an example subnet of hosts. Obtain broadcast tree for it.
6. Take a 64-bit plain 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.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17

## COMPILER DESIGN LAB (A55PC7)

**B.Tech III Year I Semester** **L/T/P/C**
**0/0/3/2**

## COURSE OBJECTIVES:
1. To provide an understanding of the language translation peculiarities by designinga complete translator for a mini language.

## COURSE OUTCOMES:
1. By this laboratory, students will understand the practical approach of now a compiler works – L2
2. This will enable him to work in the development phase of new computer languages in industry – L4

## RECOMENDED SYSTEM / SOFTWARE REQUIREMENTS:
1. Intel based desktop PC with minimum of 166 MHZ or faster processor with atleast 64 MB RAM and 100 MB free disk space
2. C++ comiler and JDK kit

Consider the following mini Language, a simple procedural high-level language, only operating on integer data, with a syntax looking vaguely like a simple C crossed with Pascal. The syntax of the language is defined by the following BNF grammar:

<program> ::= <block>
<block> ::= { <variabledefinition> <slist> } | { <slist> }
<variabledefinition> ::= int<vardeflist>;
<vardeflist> ::= <vardec> | <vardec>, <vardeflist>
<vardec> ::= <identifier> | <identifier> [ <constant> ]
<slist> ::= <statement> | <statement>; <slist>
<statement> ::= <assignment> | <ifstatement> | <whilestatement> | <block> | <printstatement> | <empty>
<assignment> ::= <identifier> = <expression> | <identifier> [ <expression> ] = <expression>
<ifstatement> ::= <bexpression> then <slist> else <slist> endif | if <bexpression> then <slist> endif
<whilestatement> ::= while <bexpression> do <slist> enddo
<printstatement> ::= print ( <expression> )
<expression> ::= <expression> <additionop> <term> | <term> | addingop> <term>
<bexpression> ::= <expression> <relop> <expression>
<relop> ::= < | <= | == | >= | > | !=
<addingop> ::= + | -
<term> ::= <term> <mulitop> <factor> | <factor>
<multop> ::= * | /
<factor> ::= <constant> | <identifier> | <identifier> [ <expression> ] | ( <expression> )
<constant> ::= <digit> | <digit> <constant>
<identifier> ::= <identifier> <letterordigit> | <letter>
<letterordigit> ::= <letter> | <digit>
<letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
<digit> ::= 0|1|2|3|4|5|6|7|8|9
<empty> has the obvious meaning

Comments (zero or more characters enclosed between the standard C / Java style comment brackets /*...*/) can be inserted. The language has rudimentary support for 1-dimensional arrays.

The declaration
int a[3] declares an array of three elements, referenced as a[0], a[1] and a[2] Note also that you should worry about the scoping of names.
A simple program written in this language is:

```
{
int a[3], t1, t2; t1 = 2;
a[0] = 1; a[1] = 2; a[t1] = 3;
t2 = -(a[2] + t1 * 6)/ a[2] - t1);
if t2 > 5 then print(t2); else
{
int t3; t3 = 99;
t2 = -25;
print(-t1 + t2 * t3); /* this is a comment on 2 lines */
}
endif
}
```

1. Design a Lexical analyzer for the above language. The lexical analyzer should ignore redundant spaces, tabs and newlines. It sholud also ignore comments. Althoughthe syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value.
2. Implement the lexical analyzer using JLex, flex or lex or other lexical analyzer generarting tools.
3. Design Predictive parser for the given language.
4. Design LALR bottom up parser for the above language.
5. Convert the BNF rules into Yacc from and write code to generate abstract syntax tree.
6. Write program to generate machine code from the abstract syntax tree generated bythe parser.

The following instruction set may be considered as target code.
The following is a simple register-based machine, supporting a total of 17 instructions. It has three distinct internal storage areas. The first is the set of 8 registers, used by the individual instructions as detailed below, the second is an area used for the storage of variables and the third is an area used for the storage of program. The instructions can be precede by a label. This consists of an integer in the range 1 to 9999 and the label is followed by a colon to seperate it from the rest of the instruction. The numerical label can be used as the argument to a jump instruction, as detailed below.
In the description of the individual instructions below, instruction argument types are specified as follows:
R specifies a register in the form R0, R1, R2, R3, R4, R5, R6 or R7 (or r0, r1, etc). L specifies a numerical label (in the rabge 1 tp 9999).
V specifies a "variable location" ( a variable number, or a variable location pointed to by a register - see below).
A specifies a constant value, a variable location, a register or a variable location pointed to by a register (an indirect address). Constant values are specified as an integer value, optionally preceded by a minus sign, preceded by a # symbol. An indirect address is specified by an @ followed by a register.
So, for example an A-type argument could have the form 4 (variable number 4), #4 (the constant value 4), r4 (register 4) or @r4 (the contents of register 4 identifies the variable location to be accessed).
The instruction set is defined as follows: LOAD A, R
loads the integer value secified by A into register R. STORE R, V
stores the value in register R to variable V. OUT R
outputs the value in register R. NEG R
negates the value in register R. ADD A, R

adds the value specified by A to register R, leaving the result in register R. SUB A, R
subtracts the value specified by A from register R, leaving the result in register R. MUL A, R
multiplies the value specified by A by register R, leaving the result in register R. DIV A, R
divides register R by the value specified by A, leaving the result in register R. JMP L
causes an unconditional jump to the instruction with the label L. JEQ R, L
jumps to the instruction with the label L if the value in register R iszero. JNE R, L
jumps to the instruction with the label L if the value in register R is notzero. JGE R, L
jumps to the instruction with the label L if the value in register R is greater than or equal to zero. JGT R, L
jumps to the instruction with the label L if the value in register R is greater than zero. JLE R, L
jumps to the instruction with the label L if the value in register R is less than or equal to zero. JLT R, L
jumos to the instruction with the label L if the value in register R is less than zero. NOP
is an instruction with no effect. It can be tagged by a label. STOP
stops execution of the machine. All programs should terminate by executing a STOP instruction.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**
**B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17**

## ADVANCED COMMUNICATION SKILLS LAB (A55HS8)

**B.Tech III Year I Semester** **L/T/P/C**
**0/0/3/2**

## INTRODUCTION
A course on Advanced Communication Skills (ACS) Lab is considered essential at the third year level of B.Tech and B.Pharmacy courses. At this stage, the students need to prepare themselves for their career which requires them to listen to, read, speak and write in English both for their professional and interpersonal communication. The main purpose of this course is to prepare the students of Engineering for their placements.

## COURSE OBJECTIVES:
1. To improve students' fluency in spoken English
2. To enable them to listen to English spoken at normal conversational speed To help students develop their vocabulary
3. To read and comprehend texts in different contexts
4. To communicate their ideas relevantly and coherently in writing To make students industry-ready
5. To help students acquire 16ehavioural skills for their personal and professional life To respond appropriately in different socio-cultural and professional contexts
6. To sensitize the importance of Soft Skills and people skills

## COURSE OUTCOMES:
After completion of the course the students will be able to Acquire vocabulary and use it contextually
Listen and speak effectively
Develop proficiency in academic reading and writing Increase possibilities of job prospects
Communicate confidently in formal and informal contexts Develop interpersonal communication skills

1. **Inter-personal Communication and Building Vocabulary** – Starting a Conversation – Responding Appropriately and Relevantly – Using Appropriate Body Language – Role Play in Different Situations – Synonyms and Antonyms, One-word Substitutes, Prefixes and Suffixes, Idioms and Phrases and Collocations.
2. **Reading Skills and Group Discussion**–General Vs Local Comprehension, Reading for Facts, Guessing Meanings from Context, Skimming, Scanning, Inferring Meaning and practice with different texts.
3. **Writing Skills** – Structure and Presentation of Different Types of Writing – Letter writing / Resume Writing/ e-correspondence/statement of purpose/ Technical Report Writing/Styles-Types-Report in Manuscript format.
4. **Group Discussion and Presentation Skills**

Group Discussions-Dynamics of Group Discussion, Intervention, Summarizing, Modulation of Voice, Body Language, Relevance, Fluency and Organization of Ideas and Rubrics of Evaluation

Concept and Process

5. **Presentation Skills** – Oral Presentations (individual or group) through JAM Sessions/ Seminars/ PPTs and Written Presentations through Posters/Projects/Reports/ emails/Assignment.
6. **Interview Skills** – Pre-interview Planning, Opening Strategies, Answering Strategies, Interview through Tele-conference & Video-conference and Mock Interviews.

**MINIMUM HARDWARE REQUIREMENT**

**Advanced English Communication Skills (AECS) Laboratory shall have the following infrastructural facilities to accommodate at least 35 students in the lab**:

Spacious room with appropriate acoustics
Eight round tables with five movable chairs for each table. Audio-visual aids
LCD Projector
Public Address system
Computer with suitable configuration
Suggested Software: The software consisting of the prescribed topics elaborated above should be procured and used.
Oxford Advanced Learner's Compass, 8th Edition
DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.

## REFERENCES:


1. Rizvi, M Ashraf. Effective Technical Communication. Mc Graw – Hill
2. Kumar, Sanjay and Pushp Lata. English for Effective Communication, OUP,2015
3. Konar, Nira. English Language Laboratories – A Comprehensive Manual, PHI Learning Pvt. Ltd,2011.
4. Shiv Khera, You can Win, Macmillan Books, New York, 2003.
5. Jeff Butterfield, Soft Skills for Everyone, Cengage Learning, 2015

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17

## WEB TECHNOLOGIES (A56PC1)

**B.Tech III Year II Semester** **L/T/P/C 4/1/0/4**

## COURSE OBJECTIVE:

Understand the concepts of PHP Language, processing of XML data with Java, Server Side Programming with Java Servlets and JSP.

## COURSE OUTCOMES:

After completion of the course the student will be able to

1. Understand basics of server side scripting using PHP – L2
2. Illustrate well formed XML programs and how to parse, use XML data with JAVA – L3
3. Design server side programming applications with servlets– L5
4. Develop programs using JSP for various applications –L5
5. Write programs with knowledge of client side scripting , validation of forms and AJAX programs – L4

## UNIT I:

**INTRODUCTION TO PHP**

Declaring variable, data variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls, like text box, 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 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 Java Script, Java Script 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 Holznera, Tata McGraw- Hill.


## REFERENCE BOOKS:


1. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dreamtech.
2. Java Server Pages: Hans Bergsten, SPD O'Reilly.
3. Programming World Wide Web, R.W. Sebesta Fourth edition, Pearson.
4. Internet and World Wide Web: How to Program, by Dietel and Nieto, Pearson.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**B.TECH. COMPUTER SCIENCE AND ENGINEERING - R17**

**DATAWAREHOUSING AND DATA MINING (A56PC2)**

**B.Tech III Year II Semester**

**L/T/P/C**
**4/1/0/4**

## COURSE OBJECTIVE:

To learn the approaches designed for handling large data with reference to data models.

## COURSE OUTCOMES:

After Completion of Course the student will be able to

1. Understand efficient and cost effective methods for maintaining data warehouse – L2
2. Apply the functionalities of Association rule mining for a given task –L3
3. Apply & Analyze classification , prediction and their respective performance evaluation metrics on sample datasets – L4
4. Illustrate ,apply and compare various clustering algorithms – L4
5. Design the required architectural schema for datamining applications – L5

## UNIT I:

### INTRODUCTION TO DATAWAREHOUSE

What is Data Warehouse Data Pre-processing, Multi-dimensional Data Model, A Three-tier Data Warehouse Architecture Efficient Computation of Data Cubes, Metadata Repository, Types of OLAP Servers, Indexing OLAP Data, and from Data Warehousing to Data Mining.

### DATA CUBE COMPUTATION AND DATA GENERALIZATION

A Road Map for the materialization of different kinds of cubes, star-cubing, Attribute oriented Induction for Data Characterization.

## UNIT II:

### MINING FREQUENT PATTERNS, ASSOCIATIONS, AND CORRELATIONS

Basic concepts a Road Map, Efficient and Scalable Frequent itemset Mining methods, Mining Various kinds of Association rules. From Strong Association Mining to Correlation Analysis.

## UNIT III:

### CLASSIFICATION AND PREDICTION

What is Classification, What is Prediction, Issues regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Classification by Back-propagation, Other Classification Methods.

### PREDICTION

Linear Regression, Non-linear Regression, other Regression-based methods, Accuracy and Error Measures, Evaluating the accuracy of classifier or a predictor.

## UNIT IV:

### CLUSTERING

What is Cluster Analysis, Types of Data in Cluster Analysis, Categorization of Major Cluster Methods, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid based Methods, Cluster High-Dimensional Data Outlier Analysis.

## UNIT V:

### APPLICATIONS AND TRENDS IN DATA MINING

Data Mining Applications, Data Mining System Products and Research Prototypes, Additional Themes of Data Mining, Social Impacts of Data Mining, Trends in Data Mining.

## TEXT BOOK:

1. Data Mining Concepts and Techniques by Jiawei Han and Micheline Kamber second edition ELSEVIER.

## REFERNCES:


1. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press.
2. Data Warehousing Fundementals, Pualraj Ponnaiah, Wiley Student Edition.
3. The Data Warehouse Life Cycle toolkit- Ralph Kimball, Wiley student Edition.
4. Data Mining, Vikaram Pudi, P. Radhakrishna, Oxford University Press.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING

## OBJECT ORIENTED ANALYSIS AND DESIGN (A56PC3)-R17

**B.Tech III Year II Semester** **L/T/P/C 4/0/0/4**

## COURSE OBJECTIVE:
Learn the paradigms of Object Oriented Approach and relating the phenomenon to a Programming Skelton.

## COURSE OUTCOMES:
After completion of the course the student will be able to
1. Illustrate the activities in the different phases of object oriented development cycle using structured approaches – L2
2. Describe how to use basic structural modelling concepts for a real world application – L2
3. Practice different system behavioral modelling techniques –L3
4. Design suitable patterns and frameworks while designing a system –L5
5. Develop different applications using their patterns and frameworks.-L5

## UNIT I:
**INTRODUCTION TO UML**

Importance of modelling, principles of modelling, object oriented modelling, conceptual model of the UML, Architecture, and 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**

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.

## TEXT BOOKS

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


## REFERENCE BOOKS

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML Pearson Education.
2. Pascal Rogues: Modelling Software Systems Using UML2, WILEY- Dream techIndia Pvt. Ltd.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING - R17

## INTRODUCTION TO ANALYTICS (A56PE4)

**B.Tech III Year II Semester** **L/T/P/C**
**3/0/0/3**

## COURSE OBJECTIVE:
To learn basic contents required for a job profile of Associate Analyst.

## COURSE OUTCOMES:
After completion of the course the student will be able to
1. Demonstrate R studio environment and understand various datatypes and datasets – L2
2. Apply various random variables by revisiting probability theorem – L3
3. Interpret SQL using R and connecting R to NoSQL databases –L3
4. Analyze data using correlation and regressive analysis – L4
5. Understand the engineering and gathering all the data related to business objective – L2

## UNIT I:
**INTRODUCTION TO ANALYTICS AND R PROGRAMMING**

Introduction to R, Rstudio (GUI): R windows Environment, introduction to various data types, Numeric character, date, data frame, array, matrix etc., Reading Datasets, Working with different file types .txt, .csvetc .outliers, Combining Datasets, R functions and Loops

**Manage Your Work to Meet Requirements**

Understanding Learning Objectives, Introduction to work & meeting requirements, Time Management, Work Management and Prioritization, Quality & Standards Adherence

## UNIT II:
**SUMMARIZING DATA & REVISITING PROBABILITY-**

Summary Statistics- Summarizing data with R, Probability, Expected, Random, Bivariate Random Variables, Probability Distribution, Central Limit Therom etc.

**WORK EFFECTIVELY WITH COLLEAGUES- INTRODUCTION**

To Work Effectively, Team Work, Professionalism, Effective Communication Skills

## UNIT III:
**SQL USING R**

Introduction to NoSQL, Connecting R to NoSQL Databases. Excel and R integration with R Connector.

## UNIT IV
**CORRELATION AND REGRESSIVE ANALYSIS**

Regression Analysis, Assumptions of OLS Regression, Regression Modelling Correlation, ANOVA, Forecasting, Heteroscedasticity, Autocorrelation, Introduction to Multiple Regression etc.

## UNIT V:
**UNDERSTAND THE VERTICALS- ENGINEERING, FINANCIAL AND OTHERS-**

Understanding Systems viz. Engineering Design, Manufacturing, Smart utilities, Production lines, Automotive, Technology etc, Understanding Business Problems related to various businesses Requirement Gathering- Gathering all the Data Related to Business Objective

## TEXT BOOKS:
1. Student Handbook for Associative Analytics

## REFERENCE BOOKS:


1. Introduction to Probability and Statistics using R, Is a textbook written for an undergraduate course in probability and statistics
2. An Introduction to R by, Venebles and Smith and the R development core team. This may be downloaded for free from the R project website (http://www.r-project.org/, see manuals).
3. Montgomery, Douglas C., and George C. Roger, Applied Statistics and Probability for Engineers. John Wiley& sons, 2010
4. Time Series Analysis and Mining with R, Yanchang Zhao.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17

## INFORMATION SECURITY MANAGEMENT (A56PE4)

**B.Tech III Year II Semester** **L/TP/C**
**3/0/0/3**

**COURSE OBJECTIVE:**
To learn basic contents required for a job profile of a Security Analyst.

**COURSE OUTCOMES:**
After completion of the course the student will be able to

1. Understanding the various fundamentals of information security- L2
2. Apply various elements of network for finding information states – L3
3. Relate statistics in terms of data leakage threats and overcome by reducing risk of data loss – L3
4. Interpret information security policies, procedures and audits – L4
5. Categorize various roles and responsibilities of information security management – L5

UNIT I:

**INFORMATION SECURITY MANAGEMENT**

Information Security Overview, Threats And Attack Vectors, Types of Attacks, Common Vulnerabilities and Exposures (CVE), Security Attacks, Fundamentals of Information Security, Computer Security Concerns, Information Security Measures etc.,

**UNIT II:**

**FUNDAMENTALS OF INFORMATION SECURITY**

Key Elements of Network, Logical Elements of Network, Critical Information Characteristics, Information states.

**UNIT III:**

**DATA LEAKAGE**

What is Data Leakage and Statistics, Data Leakage Threats, Reducing the Risk of Data Loss, Key Performance Indicators (KPI), Database Security etc.

**UNIT IV:**

**INFORMATION SECURITY POLICIES, PROCEDURES AND AUDITS**

Information security Policies-necessity- key Elements and characteristics, Security Policy Implementation, Configuration, Security Standards- Guidelines & Frame Works Etc.

**UNIT V:**

**Information Security Management Roles and Responsibilities:**

Security Roles and Responsibilities, Accountability, Roles and Responsibilities of Information Security management, team-responding to emergency situation- risk analysis process etc.

**TEXT BOOK:**

1. Management of Information Security by Michael E.Whitman and Herbert J.Mattord

**REFERENCES:**

1. http://www.iso.org/iso/home/standards/management-standards/iso2007.htm
2. http://www.csrc.nist.gov?publications/nistpubs/800-55-Rev1/SP800-55-Rev1.pdf

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17

## CLOUD COMPUTING (A56PE4)

**B.Tech III Year II Semester**

**COURSE OBJECTIVE:**

To introduce the basic concepts of cloud computing and understand the different services provided.

**COURSE OUTCOMES:**

After completion of the course the student will be able to

1. Discuss of various distributed system models and enabling technologies and concepts of virtualization – L2
2. Understand the paradigms of cloud computing – L2
3. Use various services to enhance cloud computing environment – L3
4. Construct content delivery networks clouds, resource cloud mashups - L3
5. Create data security in the cloud, to achieve production reediness for – L5

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

## UNIT I:

**SYSTEMS MODELING, CLUSTERING AND VIRTUALIZATION**

Distributed System Models and Enabling Technologies. Computer Clusters for Scalable Parallel Computing. Virtual Machines and Virtualization of Clusters and Data centres.

## UNIT II:

**FOUNDATIONS**

Introduction to Cloud Computing, Migrating into a Cloud, Enriching the 'Integration as a Service' Paradigm for the Cloud Era. The Enterprise Cloud Computing Paradigm.

## UNIT III:

**INFRASTRUCTURE AS A SERVICE (IAAS) & PLATFORM AND SOFTWARE AS A SERVICE (PAAS / SAAS)**

Virtual machines provisioning and Migration services, On the Management of Virtual machines for Cloud Infrastructures, Enhancing Cloud Computing Environments using a cluster as a Service. Secure Distributed Data Storage in Cloud Computing. Aneka, Comet Cloud, T-Systems', Workflow Engine for Clouds. Understanding Scientific Applications for Cloud Environments.

## UNIT IV:

**MONITORING, MANAGEMENT AND APPLICATIONS**

An Architecture for Federated Cloud Computing, SLA Management in Cloud Computing, Performance Production for HPC on Clouds, Best Practices in Architecture Cloud Applications in the AWS cloud, Building Content Delivery networks Clouds, Resource Cloud Mashups.

## UNIT V:

**GOVERNANCE AND CASE STUDIES**

Organisational Readiness and Change management in the Cloud age. Data Security in the Cloud, Legal issues in Cloud computing. Achieving Production Readiness for Cloud Services

## TEXT BOOKS:


1. Cloud Computing: Principles and Paradigms by Rajkumar Bi. a,
2. Distributed and Cloud Computing. Kal Hwang. Geoffeiy C.Fox. Jack J.Dongarra. Elsevier. 2012.


## REFERENCE BOOKS:


1. Cloud Computing: A Practical Approach. Anthony T.Velte. Toby J.VeFte, Robert Elsenpeter. Tata McGraw Hill. rp2Oll.
2. Enterprise Cloud Computing Gautam Shroif, Cambridge University Press. 2010.
3. Cloud Computing: Implementation, Management and Security, John W. Rittinouse, James F

Ransome. CRC Press, rp2012.
4. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. George Reese, SPD, rp2Oll.
5. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, SubraKtriaraswamy, ShahedLatif, O'Reily SPD, rp2Oll.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17

## SOFTWARE TESTING METHODOLOGIES (A56PE5)

**B.Tech III Year II Semester** **L/T/P/C**
**3/0/0/3**

## COURSE OBJECTIVE:
To learn the methodologies like flow graphs and path testing, transaction flow testing data flow testing, domain testing and logic base testing adapted in a Software Testing Process.

## COURSE OUTCOMES:
After completion of the course the student will be able to
1. Understand the purpose of testing and taxonomy of bugs, explaining flow graphs and path testing process – L2
2. Explain the process involved in testing transaction flow and data flow scenarios – L2
3. Classify and compare domain testing – L2
4. Illustrate regular expression and flow anomaly detection –L3
5. Develop graph matrices and its applications –L4

## UNIT I:
**INTRODUCTION**
Purpose of Testing, Dichotomies, Model for testing, consequences of bugs, taxonomy of bugs,
**FLOW GRAPHS AND PATH TESTING**
Basic concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

## UNIT II:
**TRANSACTION FLOW TESTING**
Transactions flows, transaction flow testing techniques.
**DATA FLOW TESTING**
Basics of Data flow testing, strategies in data flow testing, application of dataflow testing.

## UNIT III:
**DOMAIN TESTING**
Domains and paths, Nice & Ugly domains, domain testing, domains and interfaces testing, domain and testability.

## UNIT IV:
**PATHS, PATH PRODUCTS AND REGULAR EXPRESSIONS**
Path products & path expression, reduction procedure, applications, regular expressions and flow anomaly detection.
**LOGIC BASED TESTING**
Overview, decision tables, path expressions.

## UNIT V:
**STATE, STATE GRAPHS AND TRANSITION TESTING**
State graphs, good and bad state graphs, state testing, testability tips.
**GRAPH MATRICES AND APPLICATIONS**
Motivational Overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (exposure to tools like Test Director or Bad Boy)

## TEXT BOOKS:


1. Software Testing Techniques Boris Beizer, Dreamtech, Second Edition.
2. Software Testing Tools- Dr. K.V.K.K.Prasad, Dreamtech.


## REFERENCES:


1. The craft of software testing- Brain Marick, Pearson Education.
2. Introduction to Software Testing: P. Ammam & J.Offutt. Cambridge Univ. Press.
3. Software Testing M.G.Limaye TMH
4. Foundations of Software Testing, D. Grahm & Others, Cengage Learning.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING

## ARTIFICIAL INTELLIGENCE (A56PE5)-R17

**B.Tech III Year II Semester** **L/T/P/C**
**3/0/0/3**

## COURSE OBJECTIVE:
To learn and develop an ability, to formulate efficient problem space for a problem expressed in English, select a search algorithm for a problem and characterize its time and space complexities.

## COURSE OUTCOMES:
After completion of the course, the student will be able
1. Interpret the notation of state space representations and search techniques – L3
2. Interpret different logic programming concepts and logical representation techniques –L3
3. Compare and contrast between the approaches of knowledge representations –L4
4. Understand uncertainty measure probability theory – L2
5. To apply AI techniques in solving problems inferred from tasks like game-playing, Expert Systems, Machine Learning, and Natural Language Processing – L4

## UNIT I:
### INTRODUCTION
Introduction, History, Intelligent Systems, Foundations of AI, Sub areas of AI, applications. Problem Solving- State-Space Search and control strategies, introduction general problem solving, characteristics of problem, exhaustive searches, heuristic search techniques, iterative-Deepening A, Constraint Satisfaction. Game Playing, Bounded Look ahead Strategy and use of Evaluation Functions, Alpha-Beta Pruning.

## UNIT II:
### LOGIC CONCEPTS AND LOGIC PROGRAMMING
Introduction, propositional Calculus, Propositional Logic, Natural Deduction System, Axiomatic System, Semantic Tableau System in Propositional Logic, Resolution Refutation in Propositional Logic, Predicate Logic, Logic Programming.

### KNOWLEDGE REPRESENTATION
Introduction, Approaches to Knowledge Representation, Knowledge Representation using Semantic Network, Extended Semantic Networks for Knowledge Representation, Knowledge Representation using Frames.

## UNIT III:
### EXPERT SYSTEM AND APPLICATIONS
Introduction, Phases in Building Expert Systems, Expert System Architecture, Expert Systems Vs Traditional Systems, Truth Maintenance Systems, Application of Expert Systems, List of Shells and Tools.

### UNCERTAINITY MEASURE-PROBABILITY THEORY
Introduction, Probability Theory, Bayesian Belief Networks, Certainty Factor Theory, Dempster-Shafer Theory.

## UNIT IV:
### MACHINE LEARNING PARADIGMS
Introduction, Machine Learning Systems. Supervised and Unsupervised Learning, Inductive Learning. Learning Decision Trees, Deductive Learning, Cluster Support Vector Machines.

**ARTIFICIAL NEURAL NETWORKS**

Introduction, Artificial Neural Networks, Single-Layered Feed-Forward Networks, Multi-Layered Feed-Forward Networks, Radial-Basis Function Networks, Design issues of Artificial Neural Networks, Recurrent Networks.

## UNIT V:

**ADVANCE KNOWLEDGE REPRESENTATION TECHNIQUES**

Case Grammars, Semantic Web,

**NATURAL LANGUAGE PROCESSING**

Introduction, Sentence Analysis, Phases Grammars and parsers, Types of Parsers, Semantic Analysis, Universal Networking Knowledge.

## TEXT BOOKS:


1. Saroj Kaushik Artificial Intelligence Cengage Learning 2011.
2. Russell Norvig: Artificial Intelligence A Modern Approach, Pearson Education, Second Edition 2004.


## REFERNCE BOOKS:


1. Rich and Knight Artificial Intelligence, Tata McGraw Hill, Third Edition 2009
2. Introduction to Artificial Intelligence by Eugene Charniak Pearson.
3. Introduction to Artificial Intelligence and Expert Systems Dan W. Patterson. PHI
4. Artificial Intelligence by George Fluger Pearson fifth edition.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING

## SEMANTIC WEB AND SOCIAL NETWORKS (A56PE5)-R17

**B.Tech III Year II Semester** **L/T/P/C**
**3/0/0/3**

## COURSE OBJECTIVE:
To understand the knowledge representation for the semantic web, concepts of Ontology Engineering, semantic web applications, services and technology, social network analysis, and semantic web.

## COURSE OUTCOMES:
After completion of the course the student will be able
1. Understand the concepts of semantic web – L2
2. To construct and use ontologies – L3
3. Understand the concepts of semantic web and represent the required knowledge for demonstrating semantic web – L2
4. Develop the social network analysis – L5
5. Develop blog and semantic web applications – L5

## UNIT I:
Thinking and Intelligent web applications, the information age, the World Wide Web, limitations of todays web. The next generation web. Machine intelligence, Artificial Intelligence, Ontology, Inference Engines, Software Agents, Berners-Lee www, Semantic Road Map, Logic on the semantic web.

## UNIT II:
Ontologies and their role in the semantic web, Ontologies Languages for the semantic web-Resource Description Framework (RDF)/RDF Schema, Ontology Web Language (OWL), UML, XML/XML Schema. Ontology Engineering, Constructing Ontology, Ontology Development tools, Ontology Methods, Ontology sharing and merging ontology libraries and mapping.

## UNIT III:
Logic rule and inference engines, Semantic Web applications and services, semantic search, e-learning, semantic Bioinformatics, Knowledge Base.

## UNIT IV:
XML Based Web Services, Creating an OWL-S Ontology for Web Services, Semantic Search Technology, Web Search Agents and Semantic Methods, What is Social Network Analysis, development of the social networks analysis, electronic sources for network analysis-Electronic Discussion networks.

## UNIT V:
Blogs and Online Communities, Web Based Networks, Building Semantic Web Applications with Social network features.

## TEXT BOOKS

1. Thinking on the Web- Berners Lee, Godel and Turing, Wiley Interscience 2008.
2. Social Networks and the Semantic Web, Peter Mika, Springer 2007.

## REFERENCE BOOKS


1. Semantic Web Technologies, Trends and Research in Ontology Based Systems, J.Davies, Audi Studer, Paul Warren, John Wiley & Sons.
2. Semantic Web and Semantic Web Services- Liyang Lu Chapman and Hall/CRC Publishers, (Taylor & Francis Group).
3. Information Sharing on the Semantic Web- Heiner Stuckenschmidt, Frank VanHarmelen, Springer Publications.
4. Programming the Semantic Web, T.Segaran, C.Evans, J.Taylor, OReilly, SPD

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING -R17

## TECHNICAL SKILLS ENHANCEMENT LAB (A56PC7)
## R PROGRAMMING LAB

**B.Tech III Year II Semester** **L/T/P/C 0/1/3/2**

## COURSE OBJECTIVE:

This course takes you from having no previous experience in programming to an intermediate level in R. You will learn the basic toolkit of the data-oriented professional, and learn where and how to learn more advanced skills when needed.

## COURSE OUTCOMES

At the end of this course, you have all the computational tools to:

1. Write programing using R and other programming languages – L3
2. Apply empirical economic analysis – L3
3. Analyze online data science challenges – L4
4. Generate maps, plots based on the data –L5

## LIST OF PROGRAMS:

1. R and R studio set-up
2. R types, vectors and writing functions
3. More R types, vectorization and efficient R code
4. Importing data, time series data and Google Trends
5. A data science project – a first look
6. Data manipulation with tidyr
7. Plotting with ggplot2
8. R style guide, git integration, projects in R Studio
9. Maps, plots and Amazon's choice for a next office building data science project – more advanced.

## PYTHON PROGRAMMING LAB

## COURSE OBJECTIVES:

1. To write and execute programs in python to solve problems using data structures such as¬ arrays, lists, dictionaries, tuples.
2. To learn how to write python programs to implement various problem solving techniques using basic packages.

## COURSE OUTCOMES:

1. Write programs using fundamental programming elements of python –L3
2. Compare and contrast between predefined &user defined functions – L4
3. Apply python modules and data structures to solve various computing problems –L3

## LIST OF PROGRAMS:

1. Write a Python program to calculate number of days between two dates.
2. Write a python program to find the sum of the first n positive integers.
3. Write a Python program to check if a number is positive, negative or zero.
4. –Write a Python program Make a Simple Calculator
5. Write a Python program to compute the greatest common divisor (GCD) of two positive integers.
6. Write a Python function to calculate the factorial of a number (a non- negative integer). The

7. function should accept the number as an argument.
8. Write a Python program to get the Fibonacci series between 0 to 50 using recursion
9. Write a Python program to count and display the vowels of a given text.
10. Write a Python program for binary search
11. Write a Python program to print last n lines of a file.
12. Solve the following linear equations using scipy library
    X+3y+5z=10
    2x+5y+z=8
    2x+3y+8z=3
13. Draw a barchart with the following data using matplotlib
    Men_mean=[20,35,30,35,27]
    Women_mean=[25,32,34,20,25]
    Men_std=[2,3,4,1,2]
    Women_std=[3,5,2,3,3]

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING – R17

## CASE TOOLS AND WEB TECHNOLOGIES LAB (A56PC8)

**B.Tech III Year II Sem** **L/T/P/C**
**0/0/3/2**

## COURSE OBJECTIVE
Understand how UML supports the entire OOAD process, familiar with all phasesof OOAD and understand different software testing tools and their features

## COURSE OUTCOMES
1. Ability to understand the history cost of using and building CASE tools – L2
2. Ability to construct and evaluate hybrid CASE tools by integrating existing tools. Design the following UML diagrams for the Railway Reservation System – L5
3. Design various UML diagrams for real time applications –L5

1. Class Diagrams
2. Object Diagrams
3. Interaction Diagrams: i. Sequence Diagrams ii. Collaboration Diagrams
4. Behavioral Modeling i. Use case Diagrams
5. Activity Diagrams
6. Advanced Behavioral Modeling i. State Chart Diagrams
7. Architectural Modeling: i. Component Diagrams ii. Deployment Diagrams

## Web Technologies Lab

## COURSE OBJECTIVE:
Develop an ability to design and implement static and dynamic website, choose best technologies for solving web client/server problems, use appropriate client-side or Server-side applications.

## COURSE OUTCOMES:
Upon successful completion of this course, the students will be able to:
1. Design and implement dynamic websites with good aesthetic sense of designing and latest technical know-how's-L5
2. Create web pages using HTML and Cascading Styles sheets-L5
3. Analyze a web page and identify its elements and attributes-L4
4. Create dynamic web pages using JavaScript-L5
5. Develop web applications using PHP-L5
6. Create XML documents and XML Schema-L5
7. Understand, analyze and apply the role of languages like HTML, CSS, XML,JavaScript, PHP and protocols in the workings of the web and web applications-l2
8. Classify the good grounding of Web Application Terminologies, Internet Tools, E – Commerce and other web services

## LIST OF PROGRAMS:
1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.
2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.

3. Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.
4. Design an XML document to store information about a student in TKR engineering college affiliated to JNTUH. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
5. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
6. Write a PHP program to display a digital clock which displays the current time of theserver.
7. Write the PHP programs to do the following:
    a. Implement simple calculator operations.
    b. Find the transpose of a matrix.
    c. Multiplication of two matrices
8. Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas".
9. Write a PHP program that does the following:
    a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named states List.
    b. Search for a word in states that begins with k and ends in s. Perform a case in sensitive comparison. Store this word in element1 of states List.
    c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.
    d. Search for a word in states that ends in a. Store this word in element 3 of the list.
10. Write a PHP program to sort the student records which are stored in the database using selection sort.
11. A web application that lists all cookies stored in the browser on clicking “List Cookies” button. Add cookies if necessary.

# T K R COLLEGE OF ENGINEERING & TECHNOLOGY
**(Autonomous)**

## B.TECH. COMPUTER SCIENCE AND ENGINEERING – R17

## DATA WAREHOUSING AND DATA MINING LAB (A56PC9)

**B. Tech III Year II Sem** **L/T/P/C 0/0/3/2**

## COURSE OBJECTIVES:

Learn how to build a data warehouse and query it (using open source tools like Pentaho Data Integration and MS-SSIS

## COURSE OUTCOMES:

1. Apply data preprocessing and association rule mining on data using weka –L3
2. Analyze different classification algorithm using weka- L4
3. Compare clustering algorithm using weka- L4

**Build Data Warehouse and Explore WEKA**

1. Build a Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration tool, Pentoaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.).
   (i). Identify source tables and populate sample data
   (ii). Design multi-dimensional data models namely Star, snowflake and Fact constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, Manufacturing, Automobile, etc.).
   (iii). Write ETL scripts and implement using data warehouse tools
   (iv). Perform various OLAP operations such slice, dice, roll up, drill up and pivot (v). Explore visualization features of the tool for analysis like identifying trendsetc.
2. Explore WEKA Data Mining/Machine Learning Toolkit
   (i). Downloading and/or installation of WEKA data mining toolkit,
   (ii). Understand the features of WEKA toolkit such as Explorer, Knowledge Flowinterface, Experimenter, command-line interface.
   (iii). Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel)
   (iv). Study the arff file format
   (v). Explore the available data sets in WEKA.
   (vi). Load a data set (ex. Weather dataset, Iris dataset, etc.)
   (vii). Load each dataset and observe the following:
      i. List the attribute names and they types
      ii. Number of records in each dataset
      iii. Identify the class attribute (if any)
      iv. Plot Histogram
      v. Determine the number of records for each class.
      vi. Visualize the data in various dimensions

**Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets**

1. Explore various options available in Weka for preprocessing data and apply (like Discretization Filters, Resample filter, etc.) on each dataset
2. Load each dataset into Weka and run Aprori algorithm with different support and confidence values. Study the rules generated.
3. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated. Derive interesting insights and observe the effect of discretization in the rule generation process.

**Demonstrate performing classification on data sets**

1. Load each dataset into Weka and run Id3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.
2. Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix and derive Accuracy, F-measure, TPrate, FPrate, Precision and Recall values. Apply cross- validation strategy with various fold levels and compare the accuracy results.
3. Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained.
4. Plot RoC Curves
5. Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.

**Demonstrate performing clustering on data sets**

1. Load each dataset into Weka and run simple k-means clustering algorithm with different values of k (number of desired clusters)**.** Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
2. Explore other clustering techniques available in Weka.
3. Explore visualization features of Weka to visualize the clusters. Derive interesting insights and explain.

**Demonstrate performing Regression on data sets**

1. Load each dataset into Weka and build Linear Regression model. Study the clusters formed. Use Training set option. Interpret the regression model and derive patterns and conclusions from the regression results.
2. Use options cross-validation and percentage split and repeat running the Linear Regression Model. Observe the results and derive meaningful results.
3. Explore Simple linear regression technique that only looks at one variable.

**Resource Sites:**

1. **http://www.pentaho.com/**
2. **http://www.cs.waikato.ac.nz/ml/weka/ Outcomes:**
3. Ability to understand the various kinds of tools.
4. Demonstrate the classification, clusters and etc. in large data sets

# DATA MINING LAB

**Objectives:**

To obtain practical experience using data mining techniques on real world data sets and emphasize hands-on experience working with all real data sets.

## COURSE OUTCOMES:

1. Understand data from files and other sources –L2
2. Apply various data manipulation tasks on various datasets- L3
3. Apply data mining techniques on real time data sets - L3

**List of Sample Problems:**

Task 1: Credit Risk Assessment

**Description:**

The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient. To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.
2. Books. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.
3. Common sense. Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.
4. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

**The German Credit Data**:

Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. Credit dataset (original) Excel spreadsheet version of the German credit data. In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer!)

A few notes on the German dataset

1. DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
2. owns_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
3. foreign_worker. There are millions of these in Germany (many from Turrkey). It is very hard to get German citizenship if you were not born of German parents. • There are 20 attributes used in judging a loan applicant. The goal is the classify the applicant into one of two categories, good or bad.

**Subtasks: (Turn in your answers to the following tasks)**

1. List all the categorical (or nominal) attributes and the real-valued attributes seperately. (5 marks)
2. What attributes do you think might be crucial in making the credit assessment ? Come up with some simple rules in plain English using your selected attributes. (5 marks)
3. One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training. (10 marks)
4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy? (10 marks)

5. Is testing on the training set as you did above a good idea? Why or Why not? (10 marks)
6. One approach for solving the problem encountered in the previous question is using cross-validation? Describe what is cross-validation briefly. Train a Decision Tree again using cross-validation and report your results. Does your accuracy increase/decrease? Why? (10 marks)
7. Check to see if the data shows a bias against "foreign workers" (attribute 20), or "personal-status" (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka's GUI Explorer. Did removing these attributes have any significant effect? Discuss. (10 marks)
8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.) (10 marks)
9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifications equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)? (10 marks)
10. Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees? How does the complexity of a Decision Tree relate to the bias of the model? (10 marks)
11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain? Also, report your accuracy using the pruned model. Does your accuracy increase? (10 marks)
12. (Extra Credit): How can you convert a Decision Trees into "if-thenelse rules". Make up your own small Decision Tree consisting of 2-3 levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules. PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one ! Can you predict what attribute that might be in this dataset? One R classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a one R classifier. Rank the performance of j48, PART and one R. (10 marks)

**Task Resources:**

1. Mentor lecture on Decision Trees
2. Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)
3. Decision Trees (Source: Tan, MSU)
4. Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees) Weka resources:
5. Introduction to Weka (html version) (download ppt version)
6. Download Weka
7. Weka Tutorial
8. ARFF format

9. Using Weka from command line

**Task 2: Hospital Management System**

Data Warehouse consists Dimension Table and Fact Table. REMEMBER The following Dimension

The dimension object (Dimension):

_ Name

_ Attributes (Levels), with one primary key

_ Hierarchies

One time dimension is must. About Levels and Hierarchies

Dimension objects (dimension) consist of a set of levels and a set of hierarchies defined over those levels. The levels represent levels of aggregation. Hierarchies describe parent-child relationships among a set of levels. For example, a typical calendar dimension could contain five levels. Two hierarchies can be defined on these levels:

H1: YearL > QuarterL > MonthL > WeekL > DayL H2: YearL > WeekL > DayL

The hierarchies are described from parent to child, so that Year is the parent of Quarter, Quarter the parent of Month, and so forth. About Unique Key Constraints When you create a definition for a hierarchy, Warehouse Builder creates an identifier key for each level of the hierarchy and a unique key constraint on the lowest level (Base Level) Design a Hospital Management system data warehouse (TARGET) consists of Dimensions Patient, Medicine, Supplier, Time. Where measures are 'NO UNITS', UNIT PRICE. Assume the Relational database (SOURCE) table schemas as follows

TIME (day, month, year),

PATIENT (patient_name, Age, Address, etc.,)

MEDICINE (Medicine_Brand_name, Drug_name, Supplier, no_units, Uinit_Price, etc.,)

SUPPLIER :( Supplier_name, Medicine_Brand_name, Address, etc.,) If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably. Design the Hospital Management system data warehouse using all schemas. Give the example 4-D cube with assumption names.