

Artificial Intelligence (4:0:0)

Sub Code : MCA0426
Hrs/Week : 04
Hours : 3 Hrs

CIE : 50%
SEE : 50% SEE
Max.Marks : 100

Pre-requisite: NA

Course Outcomes:

On successful completion of the course, the students will be able to

1. Explain the fundamentals of Artificial Intelligence
2. Explain the search techniques and knowledge representation
3. Describe the Rules for knowledge representation.
4. Define the reasoning and planning
5. Explore the concept of Machine Learning and Common Sense

UNIT 1

Introduction, Problems, Problem Spaces & Search What is Artificial Intelligence. What is an AI Technique, The Level of the Model;**Problems, Problem Spaces & Search** Defining the problem as a state space search, Production Systems: Control strategies, Heuristic Search, Problem Characteristics: Is the problem decomposable? Can solution steps be ignored or undone? Is the Universe Predicate? Is a good solution Absolute or Relative? Is the solution a State or a Path? What is the role of knowledge? Does the task Require Interaction with a Person? Problem Classification, Production System Characteristics.

Self-Learning Exercise: Issues in the design of search programs

UNIT 2

Search Techniques and Knowledge Representation Heuristic Search Techniques: Generate-and-Test, Hill-Climbing Simple Hill climbing, Steepest Ascent Hill Climbing, Best-First search: OR Graphs, The A* Algorithm, Problem reduction: AND-OR Graphs, The AO* Algorithm. Constraint satisfaction, Means-ends analysis; Knowledge Representation: Representations and Mappings, Approaches to knowledge representation, Issues in Knowledge Representation: attributes, relationships among attributes, choosing granularity of Representation, Representing set of objects.

Self-Learning Exercise: Finding right structures as needed

UNIT 3

Representing knowledge using Rules Representing knowledge using Rules: Procedural Vs Declarative Knowledge, Logic Programming, Backward Reasoning: Backward chaining rule systems, Forward chaining Rule systems, Combining Forward and Backward Reasoning, Matching: Indexing, Matching with variables, Complex and Appropriate Matching.

Self-Learning Exercise: conflict resolution

UNIT 4

Statistical Reasoning and Planning Statistical Reasoning: Probability and Bayes' Theorem, Certainty and Rule-Based System, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic; Planning: Components of Planning system, Hierarchical Planning.

Self-Learning Exercise: Reactive Systems

UNIT 5

Learning and Common Sense what is learning? Rote Learning, Learning by taking advice, Learning in Problem-solving, Learning from Examples: Wintson Learning Program; Common Sense: Common Sense Onto logies: Time, Space, Materials, Memory Organization, Case Based Reasoning.

Self-Learning Exercise: Explanation based learning

UNIT 6

Artificial Immune Systems The phenomenon of Immunity, Immunity and Infection, The Innate Immune system, Adaptive Immune system, Clonal Selection, Learning, Immune Network Theory, Mapping Immune systems to practical Applications: Modeling affinities, Case studies: Case-1 Network Security, Case-2 AIS for Robot control and Navigation.

Self-Learning Exercise: Other applications

Text Book:

1. Elaine Rich, Kelvin Knight, Shiva Shankar B Nair: Artificial Intelligence, 3rd Edition, Tata McGraw Hill, 2009.

Reference Books:

1. Stuart Russel, Peter Norvig: Artificial Intelligence: A Modern Approach, 2nd Edition, Pearson Education, 2003.
2. Nils J. Nilsson: Principles of Artificial Intelligence, Elsevier, 1980.

Advanced Java Programming (4:0:0)**Sub Code : MCA0455****CIE : 50%****Hrs/Week : 04****SEE : 50%****SEE Hours : 3Hrs****Max.Marks : 100****Pre-requisite:** Programming in Java**Course Outcomes:**

The student on completion of this course will able to

1. Understand the concepts of JDBC.
2. Illustrate the basics of web applications using Servlet.
3. Discuss the concept of web applications using JSP technology.
4. Construct and develop how to write complex code using proper design- structure.
5. Implement the successor of JSP called the struts technology.
6. Adapt, build web applications with the navigational language designed to work with struts.

UNIT 1**8 Hours****JDBC** Introduction to JDBC, JDBC environment setup – core Java installation and sql or mysql installation, JDBC Driver types, Creating a simple JDBC application – steps involved, SQLException methods, JDBC data types, Simple programs using JDBC concepts**Self-Learning Exercise:** SQL DDL, DML and DCL concepts using JDBC.**UNIT 2****8 Hours****Servlets** Servlet Structure, Servlet packaging, HTML building utilities, Lifecycle, Single Thread model interface, Handling client requests: form data, HTTP request headers, simple programs, advanced programs using jdbc.**Self-Learning Exercise:** Handling cookies and session tracking.**UNIT 3****8 Hours****JSP** Overview of JSP technology, need of jsp, benefits of JSP, Advantages of JSP, Basic syntax, Invoking java code with JSP scripting elements, creating Template Text, Invoking java code from jsp, limited java code in jsp, using jsp expressions, comparing servlets and jsp, writing scriptlets. For example using scriptlets to make parts of

jspconditional, using declarations, declaration example.Simple programs, advanced programs using Jdbc.

Self-Learning Exercise: Installation of JSP pages, Tomcat configuration for JSP

UNIT 4

10 Hours

Introduction to Struts Why servlets are not dead?, The problems with Model 1, Model 2, Struts Overview, Upgrading to Struts 2. Model 2 applications, model 2 overview, model 2 with a servlet controller, model 2 with a filter dispatcher, Examples and programs. Starting with Struts The benefits of struts, How struts works, Interceptors, Struts Configuration files,

Self-Learning Exercise: A simple Struts Application, Dependency Injection.

UNIT5

8 Hours

Actions and Results Action classes, accessing resources, passing static parameters to an action, the Action Support class, results,

Self-Learning Exercise: Exception handling with Exception Mapping, Wildcard Mapping, Dynamic Method Invocation, Testing Action classes.

UNIT 6

10 Hours

OGNL The value stack, reading object stack object properties, reading context map object properties, invoking fields and methods, working with Arrays,

Self-Learning Exercise: working with lists, working with maps, JSP EL: When OGNL can't help.

Text Books:

1. Marty Hall, Larry Brown. Core Servlets and Java Server Pages. Volume 1: Core Technologies. Second Edition. (Chapters 3,4,5,10,11).
2. Budi Kurniawan, A Tutorial Struts 2 Design and Programming Second Ed., BPB Publications. (Chapters: 1,2,3,4)

Reference Books:

1. Kathy Sierra, Bert Bates - Head first servlets and JSP, 2nd edition. O'Reilly Publications.
2. Donald Brown, Chad Michael and others – Struts 2 in Action. Manning Publications

Advanced Web Programming (4:0:0)

Sub Code : MCA0456
 Hrs/Week : 04
 SEE Hours : 3Hrs

CIE : 50%
 SEE : 50%
 Max.Marks: 100

Pre-requisite: Web Technologies

Course Outcomes:

The student on completion of this course will able to

1. Identify the usage of Perl.
2. Illustrate the applications of CGI.
3. Select appropriate Perl or PHP for solving a Database problem.
4. Build web applications with Perl and PHP.
5. Explain Ruby and Ruby on Rails.
6. Discuss about web 2.0-Saas, Web services.

UNIT 1 8 Hours

Programming in Perl Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching.

Self-Learning Exercise: File input and output.

UNIT 2 8 Hours

CGI Scripting: What is CGI? Developing CGI Applications, Processing CGI, Introduction to CGI.pm, CGI.pm methods, Creating HTML Pages Dynamically Using CGI.pm – An Example, **Building Web Applications with Perl:** Uploading files, Using Relational Databases, using lib www

Self-Learning Exercise: Carp, Cookies.

UNIT 3 8 Hours

Introduction to PHP: Origins and uses of PHP, Overview of PHP, General syntactic characteristics, Primitives, operations and expressions, Output, Control statements Arrays, Functions, Pattern matching, Form handling.

Self-Learning Exercise: Files.

UNIT 4 10 Hours

Building Web applications with PHP: Tracking users, cookies, sessions, Using Databases. **Introduction to Ruby:** Origins and uses of Ruby, Scalar types and their

operations, Simple input and output.

Self-Learning Exercise: Control statements.

UNIT 5

8 Hours

Ruby: Arrays, Hashes, Methods, Classes, Code blocks and iterations, Pattern matching.
Introduction to Rails: Overview of Rails, Document requests processing forms.

Self-Learning Exercise: Rails applications with Databases, Layouts.

UNIT 6

10 Hours

Introduction web 2.0: What is Web 2.0?, Folksonomies and Web 2.0, Software As a Service (SaaS), Data and Web 2.0, Convergence, Iterative development, Rich User experience, Multiple Delivery Channels, Social Networking .**Web Services:** Web Services: SOAP, RPC Style SOAP, Document style SOAP, WSDL, REST services, JSON format, What is JSON?, Array literals, Object literals, Mixing literals, JSON 005Syntax.

Self-Learning Exercise: JSON Encoding and Decoding, JSON versus XML.

Text Books:

1. Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2006 (Chapter 10, 11, 13)
2. Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson Education, 2008. (Chapters 8, 11, 13, 14, 15)
3. Francis Shanahan: Mashups, Wiley India 2007 (Chapters 1, 6)

Reference Books:

1. M. Deitel, P.J. Deitel, A. B. Goldberg: Internet & World Wide Web How to H program, 3rd Edition, Pearson Education / PHI, 2004.
2. XueBai et al: The Web Warrior Guide to Web Programming, Thomson, 2003.

3. Tom White, “Hadoop: The Definitive Guide”, 3rd Edition, O’reilly, 2012.

Introduction to Android Programming

Sub Code : MCA0410

Hrs/Week : 04

SEE Hours : 3 Hrs

Pre-requisite:Java Programming

CIE : 50%

SEE : 50%

Max Marks : 100

Course Outcomes:

On completion of this course students will be able to

1. Understand the basic versions, the evolution of Android as a Mobile OS, create simple apps and apply different styles
2. Explain the User Interface of Android.
3. Manipulate the UI with different common elements.
4. Apply the knowledge to a real time Internet download app.
5. Describe persist data using preferences.
6. Apply the knowledge of DBMS to persist data.

UNIT 1

Getting Started With Android Programming: What Is Android? Android Versions, Features of Android, Architecture of Android, Android Devices in the Market, The Android Market, The Android Developer Community, Obtaining the Required Tools, Android SDK, Installing the Android SDK Tools, Configuring the Android SDK Manager, Eclipse, Android Development Tools (ADT), Creating Android Virtual Devices (AVDs), Creating Your First Android Application, Anatomy of an Android Application **ACTIVITIES, FRAGMENTS AND INTENTS:** Understanding Activities, Applying Styles and Themes to an Activity, Hiding the Activity Title, Displaying a Dialog Window, Displaying a Progress Dialog, Displaying a More Sophisticated Progress Dialog, Linking Activities Using Intents, Resolving Intent Filter, Collision, Returning Results from an Intent, Passing Data Using an Intent Object, Fragments, Adding Fragments Dynamically, Life Cycle of a Fragment, Interactions between Fragments, Calling Built-In Applications Using Intents.

Self-Learning Exercise: Understanding the Intent Object, Using Intent Filters, Adding Categories, Displaying Notifications

UNIT 2

Getting To Know The Android User Interface: Understanding the Components of a Screen, Views and ViewGroups, LinearLayout, AbsoluteLayout, TableLayout, RelativeLayout, FrameLayout, ScrollView, Adapting to Display Orientation, Anchoring Views, Resizing and Repositioning, Managing Changes to Screen Orientation, Persisting

State Information during Changes in Configuration, Detecting Orientation Changes, Controlling the Orientation of the Activity, Utilizing the Action Bar, Adding Action Items to the Action Bar, Customizing the Action Items and Application Icon, Creating the User Interface Programmatically, Listening for UI Notifications. **Designing Your User Interface With Views:** Using Basic Views, TextView, View, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views, ProgressBar View, AutoCompleteTextView, View Using Picker Views, TimePicker View, DatePicker View, Using List Views to Display Long List, ListView View, Using the Spinner View, Understanding Specialized Fragments, Using a ListFragment, Using a DialogFragment.

Self-Learning Exercise: Overriding Methods Defined in an Activity, Registering Events for Views, Using a PreferenceFragment.

UNIT 3

Displaying Pictures And Menus With Views: Using Image Views to Display Pictures, Gallery and ImageView Views, ImageSwitcher, GridView, Using Menus with Views, Creating the Helper Methods, Options Menu, Context Menu.

Self-Learning Exercise: Some Additional Views, AnalogClock and DigitalClock Views, WebView

UNIT 4

Using Internet Resources: Downloading and Parsing Internet Resources, Connecting to an Internet Resource, Parsing XML Using the XML Pull Parser, Creating an Earthquake Viewer, Using the Download Manager, Downloading Files, Customizing Download Manager Notifications, Specifying a Download Location, Cancelling and Removing Downloads, Querying the Download Manager, Using Internet Services.

Self-learning Exercise: Connecting to Google App Engine, Best Practices for Downloading Data Without Draining the Battery

UNIT 5

Files, Saving State, And Preferences: Saving Simple Application Data, Creating and Saving Shared Preferences, Retrieving Shared Preferences, Creating a Settings Activity for the Earthquake Viewer, Introducing the Preference Framework and the Preference Activity, Defining a Preference Screen Layout in XML, Native Preference Controls, Using Intents to Import System Preferences into Preference Screens, Introducing the Preference Fragment, Defining the Preference Fragment Hierarchy, Using Preference Headers, Introducing the Preference Activity, Backward Compatibility and Preference Screens, Finding and Using the Shared Preferences Set by Preference Screens, Introducing On Shared Preference Change Listeners, Creating a Standard Preference Activity for the Earthquake Viewer, Persisting the Application Instance State, Saving

Activity State Using Shared Preferences, Saving and Restoring Activity Instance State, Using the Lifecycle Handlers, Saving and Restoring Fragment Instance State, Using the Lifecycle Handlers, Including Static Files as Resources, Working with the File System, File-Management Tools Using Application-Specific Folders to Store Files, Creating Private Application Files.

Self-Learning Exercise: Using the Application File Cache, Storing Publicly Readable File.

UNIT 6

Databases And Content Providers: Introducing Android Databases, SQLite Databases, Content Providers, Introducing SQLite, Content Values and Cursors, Working with SQLite Databases, Introducing the SQLiteOpenHelper, Opening and Creating Databases Without the SQLite Open Helper, Android Database Design Considerations, Querying a Database, Extracting Values from a Cursor, Adding, Updating, and Removing Rows, Inserting Rows, Updating Rows, Deleting Rows, Creating Content Providers, Registering Content Providers, Publishing Your Content Provider's URI Address, Creating the Content Provider's Database, Implementing Content Provider Queries, Content Provider Transactions, Storing Files in a Content Provider, A Skeleton Content Provider Implementation, Using Content Providers, Introducing the Content Resolver, Querying Content Providers, Querying for Content Asynchronously Using the Cursor Loader, Introducing Loaders, Using the Cursor Loader, Adding, Deleting, and Updating Content, Inserting Content, Deleting Content, Updating Content, Accessing Files Stored in Content Providers.

Self-Learning Exercise: Creating a To-Do List Database and Content Provider.

Text Books:

1. Wei-Meng Lee, Beginning Android 4 Application Development, Wrox, Wiley India Edition. (Chapters: 1, 2, 3, 4, 5)
2. Reto Meier, Professional Android 4 Application Development, Wrox, Wiley India Edition(Chapters: 6, 7, 8)

Reference Books:

1. Mark Murphy, The Busy Coder's Guide to Android Development, version 4.2.
2. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides)
3. Paul Deitel, Harvey Deitel, Abbey Deitel, and Michael Morgano, Android for Programmers: An App-Driven Approach
4. AnubhavPradhan, Anil V Deshpande, Composing Mobile Apps: Learn / Explore / Apply using Android 1stEdition.

Big Data Analytics (4:0:0)

Sub code	: MCA0408	CIE	: 50%
Hrs /week	: 04	SEE	: 50%
SEE Hrs	: 3Hrs	Max. Marks	: 100

Pre-requisite : NA

Course Outcomes:

On successful completion of the course the students will be able to

1. Define and relate several key technologies used in manipulating, storing, and analyzing big data.
2. Outline a clear understanding of R &Hadoop.
3. Experiment with Integrating R &Hadoop for solving big data problems.
4. Analyze and get a clear understanding of Hadoop Streaming and its importance.
5. Measure Big Data and analyze Big Data.
6. Design and develop applications to apply tools and techniques for analyzing Big Data.

UNIT 1

Introduction To Big Data Big Data and its Importance – Four V’s of Big Data – Drivers for Big Data –Introduction to Big Data Analytics – Big Data Analytics applications.

Self-Learning Exercise: Big Data Analytics Applications

UNIT 2

Introduction To R & Hadoop Getting Ready to Use R and Hadoop, Installing R, Installing RStudio, Understanding the features of R language, Installing Hadoop, Understanding Hadoop features, Learning the HDFS and MapReduce architecture, Writing HadoopMapReduce Programs, Introducing HadoopMapReduce, Understanding the HadoopMapReduce fundamentals, Writing a HadoopMapReduce example, Learning the different ways to write HadoopMapReduce in R

Self-Learning Exercise: Installing R Studio

UNIT 3

Integration Of R & Hadoop Integrating R and Hadoop, Introducing RHIPE ,Understanding the architecture of RHIPE Understanding RHIPE samples, Understanding the RHIPE function reference, Introducing RHadoop, Understanding the architecture of RHadoop, Understanding RHadoop examples, Understanding the RHadoop function reference.

Self-Learning Exercise: Installation of RHadoop

UNIT 4

Hadoop Streaming With R Using Hadoop Streaming with R - Introduction, Understanding the basics of Hadoop streaming, Understanding how to run Hadoop streaming with R, Understanding a MapReduce application, Exploring the Hadoop Streaming R package

Self-Learning Exercise: Hadoop Streaming Packages.

UNIT 5

Data Analytics With R And Hadoop Understanding the data analytics project life cycle – Introduction, Identifying the problem, Designing data requirement, Preprocessing data, Performing analytics over data, Visualizing data, Understanding data analytics problems, Exploring web pages categorization Case Studies: Computing the frequency of stock market change, Predicting the sale price of blue book for bulldozers.

Self-Learning Exercise: Case Study: Stock Market change

UNIT 6

Understanding Big Data Analysis With Machine Learning Introduction to machine learning, Types of machine-learning algorithms, Supervised machine-learning algorithms, Unsupervised machine learning algorithm, Recommendation algorithms, Steps to generate recommendations in R, Generating recommendations with R and Hadoop.

Self-Learning Exercise: Unsupervised machine learning algorithm

Textbooks :

1. Arvind Sathi, “Big Data Analytics: Disruptive Technologies for Changing the Game”, 1st Edition, IBM Corporation, 2012 (Chapter 1,2,3 Unit 1)
2. Big Data Analytics with R and Hadoop, Vignesh Prajapati, -Packt Publishing 2013 (Chapters 1,2,3,4,5,6 Unit 2,3,4,5,6)

Reference Books:

1. Michael Minelli, Michele Chambers, “Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today’s Business”, 1st Edition, Ambiga Dhiraj, Wiley CIO Series, 2013.
2. Bill Franks, “Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics”, 1st Edition, Wiley and SAS Business Series, 2012.

Cloud Computing (4:0:0)

Sub code : MCA0446
Hrs /week : 04
SEE Hrs : 3Hrs

CIE : 50 %
SEE : 50 %
Max. Marks : 100

Pre-requisite : NA

Course Outcomes:

On successful completion of the course the students will be able to

1. Classify different cloud computing models.
2. Describe basic cloud architecture and cloud services
3. Use the concepts of abstraction and virtualization to plan a cloud environment
4. Demonstrate cloud management using Google web services
5. Discuss cloud security issues
6. Contrast the relationship between SOA and cloud computing by moving application toCloud.

UNIT 1

Introduction Defining Cloud Computing, Cloud Types: NIST Model, Cloud Cube Model, Deployment models, Service Models, Characteristics of Cloud Computing: Paradigm shift, Benefits of Cloud computing, Disadvantages of cloud computing, Role of Open Standards, Assessing the Value Proposition: Measuring the Cloud's Value, Avoiding Capital Expenditures, Computing the Total Cost of Ownership, Specifying Service Level Agreements, Defining Licensing Models.

Self-Learning Exercise: Measuring Cloud Computing Cost

UNIT 2

Understanding Cloud Architecture Exploring the Cloud Computing Stack: Composability, Infrastructure, Platforms, Virtual Appliances, Communication Protocols, Applications, Connecting to the cloud: JoliCloud Netbook OS.

Self-Learning Exercise: Chromium OS: The Browser as an Operating System.

UNIT 3

Understanding Services, Applications, Abstraction and Virtualization Defining Infrastructure as Service (IaaS), Defining Platform as a Service (PaaS), Defining Software as a Service (SaaS), Defining Identity as a Service (IDaaS), Using Virtualization Technologies, Load Balancing and Virtualization, Understanding Hypervisors, Understanding Machine Imaging.

Self-Learning Exercise: Defining Compliance as a Service (CaaS), Porting Applications

UNIT 4

Using Google Web Services and Managing the Cloud Exploring Google Applications, Surveying the Google Application Portfolio: Indexed Search, The dark Web, Aggregation and disintermediation, Productivity applications and services, Enterprise offerings, Google Analytics, Google Translate, Administrating the Clouds, Cloud Management Products, Emerging Cloud Management Standards.

Self-Learning Exercise: Google Web Toolkit, Google App Engine

UNIT5

Understanding Cloud Security Securing the Cloud: The security boundary, Security service boundary, Security mapping. Security Data: Brokered cloud storage access, Storage location and tenancy, Encryption, Auditing and compliance, Establishing Identity and Presence: Identity protocol standards.

Self-Learning Exercise: Windows Azure identity standards, Presence.

UNIT 6

Understanding Service Oriented Architecture and Moving Applications to the Cloud Introducing Service Oriented Architecture: Event-driven SOA or SOA 2.0, Enterprise Service Bus, Service Catalogs, Defining SOA Communications: Business Process Execution Language, Business Process modeling. Managing and Monitoring SOA: SOA Management tools, SOA security, The Open Cloud Consortium, Moving Applications to the Cloud: Applications in the Clouds.

Self-Learning Exercise: Relating SOA and Cloud Computing, Applications and Cloud APIs, Development of Cloud Applications using SOA

Text Book:

1. Cloud Computing by Barrie, Wiley India, Publication year 2011

Reference Books:

1. Cloud Computing for Dummies by Judith Hurwitz, R. Bloor, M. Kanfman, F. Halper(Wiley India Edition).
2. Cloud Security by Ronald Krutz and Russell Dean Vines, Wiley-India.
3. Cloud Computing: A Practical Approach, Anthony T Velte.
4. Google Apps by Scott Granneman, Pearson A Brief Guide to Cloud Computing: An Essential Introduction to the Next Revolution in Computing, Christopher Barnatt.

Data Mining (4:0:0)

Sub Code : MCA0429
 Hrs/Week : 04
 SEE Hours : 3 Hrs

CIE : 50%
 SEE : 50%
 Max Marks : 100

Pre-requisite : NA

Course Outcomes:

On successful completion of the course, the students will be able to

1. Understand the necessity of data mining.
2. Identify different tasks of data mining.
3. Identify the different methods in association analysis
4. Apply algorithms for association analysis in identifying associations
5. Understand the importance of cluster analysis.
6. Understand the importance of advanced data mining topics like spatial data mining, text mining, mining the web.

UNIT 1

Introduction, Data – 1 What is Data Mining? Motivating Challenges; The origins of data mining; Data Mining Tasks. Types of Data.

Sel- Learning Exercise: Data Quality.

UNIT 2

Data – 2 Data Preprocessing; Measures of Similarity and Dissimilarity
Classification Preliminaries; General approach to solving a classification problem; Decision tree induction; Rule-based classifier;

Self-Learning Exercise: Nearest-neighbor classifier.

UNIT 3

Association Analysis – 1 Problem Definition; Frequent Itemset generation; Rule Generation; Compact representation of frequent itemsets;

Self -Learning Exercise: Alternative methods for generating frequent itemsets.

UNIT 4

Association Analysis – 2 FP-Growth algorithm, Evaluation of association patterns; Effect of skewed support distribution.

Self-Learning Exercise: Sequential patterns.

UNIT 5

Cluster Analysis Overview, K-means, Agglomerative hierarchical clustering, DBSCAN.

Self-Learning Exercise: Overview of Cluster Evaluation.

UNIT 6

Further Topics in Data Mining Multidimensional analysis and descriptive mining of complex data objects; Spatial data mining; Multimedia data mining; Text mining; Mining the WWW, Outlier analysis. **Applications** Data mining system products and research prototypes; Additional themes on Data mining; Social impact of Data mining; Trends in Data mining.

Self-Learning Exercise: Data mining applications

Text Books:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson Education, 2007. (Chapter 1, 2, 4.1 to 4.3, 5.1, 5.2, 6, 8.1 to 8.4, 8.5.1)
2. Jiawei Han and Micheline Kamber: Data Mining – Concepts and Techniques, 2nd Edition, Morgan Kaufmann, 2006. (Chapters 7.11, 10, 11)

Reference Books:

1. K.P.Soman, Shyam Diwakar, V.Ajay: Insight into Data Mining – Theory and Practice, PHI,

Introduction to Python Programming (4:0:0)

Sub code	: MCA0407	CIE	: 50%
Hrs /week	: 04	SEE	: 50%
SEE Hrs	: 3Hrs	Max. Marks	: 100

Pre-requisite : NA

Course Outcomes:

On successful completion of the course the students will be able to

1. Recall the need for a “programming language” like Python. Also differentiate the differences b/w Python versions.
2. Understand and demonstrate how to execute basic code in Python.
3. Apply various python data types to create and design some simple python scripts.
4. Compare and classify the complex python data types to execute python scripts.
5. Discuss and develop python scripts using the concept of basic python statements.
6. Create, design and develop scripts using python statements, loops and expressions.

UNIT 1

Python Q & A Session Why do people use Python? What can I do with Python? Intro to python: how is python developed and supported? Advantages of using python: What are python's technical strengths? How Python runs programs: Introducing the Python interpreter, Program execution, Python's view

Self-Learning Exercise: Installation of Python in Windows and Linux Environment, alternatives to python.

UNIT 2

How To Run Python Programs The interactive prompt, starting an interactive session, why the interactive prompt, System Command lines and files – a first script, Module imports and reloads, GUI to python programming and other IDEs, Other launch options. Introducing python object types – The python conceptual hierarchy, why use built in types?, Numbers, strings, lists, dictionaries, tuples, files, Other core types.

Self-Learning Exercise: Test your knowledge section.

UNIT 3

Getting Deeper Into Data Types – INumeric types – Numeric type basics, numeric literals, built in numeric tools, Numbers in action, other numeric types. The dynamic

typing interlude – the case of the missing declaration statements, shared references, dynamic typing is everywhere.

Self-Learning Exercise: Weak references, Test your knowledge section.

UNIT 4

Getting Deeper Into Data Types – Ii String fundamentals – basics, literals, string basic operations, methods, formatting expressions, formatting method calls. Lists and dictionaries – Lists in action, dictionaries in action.

Self-Learning Exercise: Some examples of strings, lists and dictionaries.

UNIT 5

Introduction To Python Statements – Introducing python statements – The python conceptual hierarchy revisited, Python's statements, a tale of two ifs, syntaxes. Assignments, expressions, and prints – assignment statement forms, sequence assignments, extended sequence unpacking in 3.x, multiple target assignments, Expression statements, python print statements, print stream redirection.

Self-Learning Exercise: Test your knowledge section.

UNIT 6

Introduction To Python Statements – Iiif Tests and syntax rules - Python syntax revisited, truth values and boolean tests, the if-else ternary expression, some programs/scripts. While and for loops – while loops, break, continue, pass and the loop else, for loops, loop coding.

Self-Learning Exercise: Largest of 2 numbers and other basic scripts.

Textbooks :

1. Mark Lutz, “Learning Python”, 5th edition, O'Reilly publications, 2013.
(Chapters – 1,2,3,4,5,6,7,8,10,11,12,13)

Reference Books:

1. Mark Lutz, “Programming Python”, 4th edition, O'Reilly publications, 2010
2. Zed A Shaw, “Learn Python the hard way”, Hard Way Series, 2013.

LINUX Systems Programming (4:0:0)

Sub Code : MCA0458
Hrs/Week : 04
SEE Hours : 3 Hrs

CIE : 50%
SEE : 50%
Max.Marks : 100

Pre-requisite:NA

Course Outcomes:

On successful completion of the course, the students will be able to

2. Identify the role of various LINUX programming standards and file types.
3. Demonstrate the usage of various system calls to perform file operations.
4. Discuss LINUX OS concepts such as: kernel support for files.
5. Illustrate process-controlling procedures, relationship between processes, process groups.
6. Describe system limits and understand the uname command.
7. Understand the various file systems.

UNIT1

History and Standards: A Brief History of UNIX and C, A Brief History of Linux, Standardization. **Fundamental Concepts:** The Core Operating System: The Kernel, The Shell, Users and Groups, Single Directory Hierarchy, Directories, Links, and Files, File I/O Model, Programs, Processes, Static and Shared Libraries, The /proc file system, Summary.

Self-learning Exercise: Pseudo-terminals, Date and Time.

UNIT 2

System Programming Concepts: System Calls, Library Functions, The Standard C Library, Handling Errors from System Calls and Library Functions, Portability Issues, **File I/O:** Overview, Universality of I/O, Opening a File: open(), Reading from a File: read(), Writing to a File: write(), Closing a File: close().

Self-learning Exercise: Changing the File Offset: lseek().

UNIT 3

File I/O: Further Details: File Control Operations: fcntl(), Open File Status Flags, Relationship Between File Descriptors and Open Files, Duplicating File Descriptors, File

I/O at a Specified Offset: pread() and pwrite(), Scatter-Gather I/O: readv() and writev(),
Truncating a File: truncate() and ftruncate().

Self-learning Exercise: Creating Temporary Files.

UNIT 4

Processes: Processes and Programs, Process ID and Parent Process ID, Memory Layout of a Process, Virtual Memory Management, Command-Line Arguments (argc, argv), Environment List, Performing a Nonlocal Goto: setjmp() and longjmp(), **Users And Groups:**The Password File: /etc/passwd, The Shadow Password File: /etc/shadow, Retrieving User and Group Information, Password Encryption and User Authentication.

Self-learning Exercise: The Group File: /etc/group.

UNIT 5

System Limits And Options: System Limits, Retrieving System Limits (and Options) at Run Time, Retrieving File-Related Limits (and Options) at Run Time, **System And Process Information:** The /proc File System, System Identification: uname().

Self-learning Exercise: Usage of uname command.

UNIT 6

File Systems: Device Special Files (Devices), File Systems, I-nodes, The Virtual File System (VFS), Journaling File Systems, Single Directory Hierarchy and Mount Points, Mounting and Unmounting File Systems, A Virtual Memory File System: tmpfs

Self-learning Exercise: Disks and Partitions.

Text Books:

1. Michael Kerrisk: The LINUX Programming Interface, 1st Edition, No Starch Press, 2010. (Chapters 1-6 and 8, 11,12,14)

Reference Books:

1. Marc J. Rochkind: Advanced UNIX Programming, 2nd Edition, Pearson Education, 2005.
2. W.Richard Stevens, Stephen A. Rago: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education / Prentice-Hall of India, 2005.

Mini Project 1 (0:0:2)

SubjectCode : MCA0120

Hrs/Week : 02

CIE : 50%

SEE : 50%

Max.Marks : 50

Pre-requisite: NA

Course Outcomes:

On successful completion of the course, the students will be able to:

- 1. Identify of real world problems**
- 2. Understand different design methodologies.**
- 3. Select appropriate tools for solving a problem**
- 4. Decide on advanced programming techniques/languages**
- 5. Apply the art of verification and validation.**
- 6. Write technical reports**

A team of TWO students must develop the mini project. However, during the final presentation, each student must demonstrate the project individually.

- The team may implement a mini project of their choice.
- The team must submit a Brief Project Report (25 to 30 Pages) that must include the following:

1. Introduction,
2. Requirements
3. Software Development Process Model Adopted, Analysis and Design Models.
4. Implementation
5. Testing

The Report must be evaluated for 10 marks, Demonstration for 30 marks and Viva for 10 mark

ELECTIVES

Seminar- II (0:0:2)MCA0201

Course Outcomes:

On successful completion of the course, the students will be able to

1. Understand current trends in specific area of interest
2. Perform literature survey related to the specific topics of interest
3. Appreciate the results of technical work
4. Write technical reports
5. Summarize and present the technical contents

A Seminar should be given by an individual student based on topics chosen from the emerging areas and technologies of Computer science & Computer Applications. References from journals such as IEEE, ACM etc., shall be used. A report on this seminar with 15-20 pages shall also be prepared.

Major Project MCA2801

Course Outcomes:

On successful completion of the course, the students will be able to

1. Analyze real world problems
2. Implement the different design methodologies
3. Select appropriate tools for solving problems
4. Implementing advanced programming techniques/languages
5. Perceive the art of verification and validation
6. Write technical reports

Major project must be carried out by individual student, one project per student.

Student must submit a **Detailed Project Report** (60 to 80 Pages) in a format as specified by the department.

Internal guides will evaluate the performance (Continuous Internal Evaluation) for 50 Marks.

The Report will be evaluated for 125 marks by both internal and external evaluators.

Final viva-voce which includes demonstration and presentation of project work will be evaluated jointly by internal and external examiners for 75 marks.

.Net Laboratory (0:0:3)MCA0113

Course Outcomes:

On successful completion of the course, the students will be able to

1. Display proficiency in C# by building stand-alone applications in the .NET framework using C# that is completely event driven.
2. Create simple web applications and window application features of C#
3. Create distributed data-driven applications using the .NET Framework, C#
4. Create web-based distributed applications using C#
5. Debug an application using breakpoints and Try/Catch/Finally blocks
6. Debug an application using delegates and demonstration of inheritance support.

IDE: Microsoft Visual Studio8 Initially work with simple programs in runtime environment .NET framework i.e., SDK command prompt.

Simple programs in C#, Programs on Array processing in C#, interfaces, use of Virtual and override key words, collections, abstract classes and methods, exception handling etc.

Mini Project MCA0114

Course Outcomes:

On successful completion of the course, the students will be able to

1. Analyze real world problems
2. Implement different design methodologies
3. Select appropriate tools for solving problems
4. Implementing advanced programming techniques/languages
5. Perceive the art of verification and validation
6. Write technical reports

Mini project must be carried out by groups of two student, one project per group.

Student must submit a **Detailed Project Report** in a format as specified by the department.

Seminar-I (0:0:2)MCA0115

Course Outcomes:

On successful completion of the course, the students will be able to

1. Understand current trends in specific area of interest
2. Perform literature survey related to the specific topics of interest
3. Appreciate the results of technical work

4. Write technical reports

5. Summarize and present the technical contents

A Seminar should be given by an individual student based on topics chosen from the emerging areas and technologies of Computer science & Computer Applications. References from journals such as IEEE, ACM etc., shall be used. A report on this seminar with 15-20 pages shall also be prepared