DATA SCIENCES

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand data pre-processing, analysis, regression techniques and software tools for data science.
2. To develop comprehensive understanding of various kind of learning.

UNIT I -Introduction to Data Science: Data Science, Data Cleaning, Data Integration and Transformation, Data Reduction, discretization and Concept Hierarchy Generation, Traits of Big data, Web Scraping, Hypothesis and inference, Analysis vs Reporting.
Data Science Toolkits using Python: Matplotlib, NumPy, Scikit-learn. Visualizing Data: Bar Charts, Line Charts, and Scatterplots. Working with data: Reading Files, Scraping the Web, Using APIs (Example: Using the Twitter APIs), Cleaning and Munging, Manipulating Data, Rescaling, Dimensionality Reduction, PCA.
Case Study: Getting Data -Twitter Data API

UNIT II - Machine Learning
Well posed learning problems -Design a learning task, issues in Machine Learning,
Concept Learning: Learning as a Task, Learning as a search, Version Space, Candidate Elimination Algorithm. Machine learning - Overfitting and Underfitting, Correctness, Feature Extraction and Selection, Types of Machine Learning.
Case Study: Working with Twitter Data

UNIT III – Regression
Introduction to Simple Linear Regression, Model Building, Estimation of Parameters using Ordinary Least Squares, Logistic Regression, Model Building, Interpretation of Logistic Regression Parameters, K-Nearest Neighbor Classifiers.

UNIT IV - AI for Data Science
Introduction to Neural networks, Biological Inspiration, Characteristics of Neural Networks, Learning Methods, Perceptron, Multilayer Network and Back propagation, Remarks of BPN Algorithm, Taxonomy of NN Architectures, random forest, Classification Errors.

UNIT V: Genetic Algorithms and Reinforcement Learning
Genetic Algorithms: Motivation, GA, Illustrative Example, Hypothesis Space Search, Genetic Programming, Models of Evolution and Learning.
Reinforcement Learning: Introduction, The Learning Task, Q learning, Non-Deterministic Rewards and Actions, Temporal Difference Learning, Generalizing from Examples, Relationship to Dynamic Programming, Overview of Deep learning.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Demonstrate understanding of the mathematical foundations needed for data science
CO 2 : Understand Machine Learning techniques.
CO 3 : Exhibit proficiency with statistical analysis of data through Machine learning
CO 4 : Apply principles of Data Science to the analysis of business problems through Neural Networks, Machine Learning, Genetic Algorithms.
CO 5 : Build data science applications using Python based toolkits.

Text Books:

1. Tom M. Mitchell, "*Machine Learning*" McGraw-Hill Science/Engineering/Math
2. Joel Grus, "*Data Science from Scratch: First Principles with Python*", O'Reilly Media
3. Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, 3rd edition, Morgan Kaufmann Publishers, 2011

References:

1. Forsyth, Probability and Statistics for Computer Science, David, Springer.
2. S.Rajasekaran and G.A.Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic, and Genetic Algorithms, Synthesis and Applications*, PHI Learning Private Limited
3. Jain V.K., "Data Sciences", Khanna Publishing House, Delhi.

INTERNET OF THINGS

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To impart necessary and practical knowledge of components of IoT
2. To demonstrate the various communication and connectivity protocols
3. To discuss the design issues of industry IoT through cloud infrastructure

Unit I –Introduction to Internet of Things
Introduction - Definition and Characteristics of IoT. Physical Design of IoT – IoT Protocols. Logical Design of IoT -IoT Communication Models, IoT Communication APIs. IoT Enabling Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems. IoT Levels and Deployment Templates.

Unit II - Python packages
JSON, XML, HTTPLib, URLLib, SMTPLib, RPi.GPIO.
IoT Physical Devices and Endpoints - Introduction to Raspberry PI, Interfaces (serial, SPI, I2C). Programming Raspberry PI with Python - Controlling LED, Interfacing an LED and Switch, Interfacing a Light Sensor with Raspberry Pi.

Unit III – Domain Specific IoTs
IoT and M2M – Introduction – M2M, Difference between IoT and M2M, SDN and NFV for IoT.
IoT System Management with NETCONF-YAN - SNMP, NETCONF, YANG, IoT System Management with NETCONF-YANG, NETOPEER.

Unit IV – IoT Physical Servers and Cloud Offerings
Cloud Computing – Definition, Characteristics, Architecture, Service Models and Deployment Models, Virtualization Concepts. Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Cloud for IoT, Python Web Application Framework, Designing a RESTful Web API.

Unit V – Case Studies Illustrating IoT Design
Introduction - Home, City, Environment, Energy Systems, Retail, Logistics, Agriculture, Industry, Health and Lifestyle.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand the characteristics, protocols and communication models required for logical design of IoT.
CO 2 : Realize the hardware platforms for implementing and interfacing the IoT based board with different peripheral devices and serial communication devices.
CO 3 : Gain knowledge on protocol stacks for IoT and M2M networks and configurations.
CO 4 : Integrate devices and develop an application that can communicate through IoT Cloud.
CO 5 : Implement various case studies in IoT design.

Text Books:

1. Arshdeep Bahga and Vijay Madisetti, *Internet of Things - A Hands-on Approach,* Universities Press, 2015.
2. Peter Waher, *Learning Internet of Things,* Packt publisher, 2015.
3. Matt Richardson & Shawn Wallace, *Getting Started with Raspberry Pi,* O'Reilly (SPD), 2014.

References:

1. Kimmokarvinen and teroKarvenien, *Getting started with sensors: Measure the world with Electronics, Arduino, and Raspberry,* First Edition, Shroff/O'Reilly, 2014.
2. Richardson Matt, *Getting started with Raspberry Pi,* Shroff Publishers & Distributers Private Limited.

ADVANCED WEB TECHNOLOGIES

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

To provide good understanding of latest web technologies on client and server-side components as follows:

1. Client Side: JavaScript, Angular 2
2. Server Side: Node, Express
3. Database: No SQL

UNIT I

Revisiting HTML 5, CSS 3. Version Control (Git). **ES 6 Features:** fat arrows, classes, destructuring, let & const, modules, symbols, function generators, async / await for handling asynchronicity.

UNIT II

No SQL:

Introduction to NoSQL, SQL vs NoSQL, Migrating from SQL to NoSQL database. Different Types of NoSQL databases, CAP Theorem, Sharding.

Introduction to MongoDB, Key features of MongoDB, MongoDB shell, MongoDB databases, MongoDB collections, MongoDB CRUD operations, Real – time database Firebase CRUD operations.

UNIT III

Getting started with Node.js:

Introduction to Node.js, REPL, Node modules: events, os, http, buffers, streams. Building own API and consuming it. (**REST full API**).

Building an Express web application:

Introduction to Express, Installation of Express, Create first Express application, application, request, and response objects, Configuring an Express application, Rendering views, Authentication, Authorization.

Type Script: compiler options, Classes, Interfaces, Generics, Decorators

UNIT IV

Introduction to Angular 2:

Introduction to Angular 2, Angular 2 Architecture, State Management, Validation, Routing. Passing data from parent to child and Passing data between siblings.

Angular 2 Specific: Directives, Modules, Components, Services, Consuming API, Observables, Binding, Types of Forms, Pipes, Dependency Injection.

UNIT V

Introduction to ReactJS: Motivation for using React, Key differentiators (Virtual DOM, One-way binding)

React Components; React Component, Render function, Component API, Component Lifecycle, State, Props, Mixins, JSX, Building a simple React Component.

Component inter communication: Component Composition, pass data from parent to child, pass data from child to parent, Fetching data API using axioms, Form Validations, Posting Data, React Router, Building & Deploying React App.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand the use of prototyping, ES6 classes, advanced JavaScript concepts and make use of the XMLHttpRequest API.

CO 2 : Define, compare and use of the four types of NoSQL Databases.

CO 3 : Design and build robust REST APIs using Node.js, demonstrate the Express framework and work with Typescript.

CO 4 : Implement one way / two way data binding for data interpolation, dependency

injection and design single page application using Angular2.

CO 5 : Create smaller components to build Interactive User interfaces using ReactJS.

Text Books:

1. Amos Q. Haviv, *MEAN Web Development*, Second Edition, Packt Publications, November 2016
2. Vasan Subramanian, "*Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node*" ,2nd Edition, APress.

References:

1. Shelly Powers, *Learning Node: Moving to the Server-Side*, 2nd Edition, O'REILLY, 2016.
2. Simon D. Holmes and Clive Harber, "*Getting MEAN with Mongo, Express, Angular, and Node*", Second Edition, Manning Publications,2019
3. Brad Dayley, "*Node.js, MongoDB and Angular Web Development*", 2nd Edition , Addison-Wesley Professional,2017
4. https://angular-2-training-book.rangle.io.
5. https://www.atlassian.com/git
6. https://www.typescriptlang.org/docs/handbook/basic-types.html
7. https://firebase.google.com

PRINCIPLES OF ARTIFICIAL INTELLIGENCE

(Professional Elective I)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To develop an understanding of the basic concepts of Artificial Intelligence.
2. To analyze the nature of various advanced search strategies in AI.
3. To impart knowledge representation, reasoning and planning mechanisms.
4. To understand the game playing and Natural Language Processing tools in AI.
5. To give an understanding about PSO and Multi-Agent Systems.

UNIT I –Problem and Search, AI Introduction, Problems, Underlying Assumption, AI Technique, Leve of the Model. Problem, Problem Spaces, and Search – Defining the Problem as a State Space Search, Production systems, Problem characteristics, Production System Characteristics, Issues in the Design of Search Programs.

UNIT II – Heuristic Search Techniques, Generate and Test, Hill Climbing, Best-first search, Problem reduction, constraint satisfaction, mean end analysis. Knowledge Representation – Issues, Representations and Mappings, Approaches to knowledge representation, Issues in Knowledge Representation, The Frame Problem. Using Predicate Logic, Representing Simple Facts in Logic, Representing Instance and ISA Relationships, Computable Functions and Predicates, Resolution.

UNIT III – Game Playing, The Minimax Search Procedure, Adding Alpha-beta cutoffs, Additional Refinements, Iterative Deepening, Specific Games. Natural Language Processing- Introduction, Syntactic Processing, Semantic Analysis, Discourse and Programmatic Processing, Statistical Natural Language Processing, Spell Checking. Perception and Action – Real time search, perception, action, Robot Architecture.

UNIT IV – Multi-Agent System, Introduction, What Are Agents?, Examples of Agents , Intelligent Agents , Agents and Objects, Agents and Expert Systems , Abstract Architectures for Intelligent Agents ,Purely Reactive Agents , Perception , Agents with State , Concrete Architectures for Intelligent Agents , Logic-based Architectures , Reactive Architectures , Belief-Desire-Intention Architectures , Layered, Motivations, Characteristics of Multi-agent Environments, Agent Communications, Coordination , Dimensions of Meaning, Message Types, Communication Levels, Speech Acts, Knowledge Query and Manipulation Language (KQML), Knowledge Interchange Format (KIF), Ontologies.

UNIT V – Particle Swarm Optimization, Basic Particle Swarm Optimization, Global Best PSO, Local Best PSO, gbest versus lbest PSO, Velocity Components, Geometric Illustration, Algorithm Aspects, Social Network Structures, Basic Variations, Velocity Clamping, Inertia Weight, Constriction Coefficient, Synchronous versus Asynchronous Updates , Velocity Models, Basic PSO Parameters, Single-Solution Particle Swarm Optimization, Guaranteed Convergence PSO, Social-Based Particle Swarm Optimization, Hybrid Algorithms, Sub-Swarm Based PSO, Multi-Start PSO Algorithms, Repelling Methods, Binary PSO.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Identify the scope for agent based engineering solutions using AI based tools.
CO 2 : Demonstrate advanced search strategies.
CO 3 : Master the knowledge representation based AI solutions.
CO 4 : Understand the role of multi-agent systems in AI.
CO 5 : Give a complete basic knowledge about PSO and AI.

Text Books:

1. Elaine Rich, Kevin Knight, Shivashankar B. Nair, *Artificial Intelligence* ,3rd Edition, McGraw-Hill, 2017.
2. *Weiss, G., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA: MIT Press, 1999. (UNIT IV)*
3. Andries P. Engelbrecht, *Computational Intelligence: an Introduction*, Second Edition, John Wiley & Sons, 2007 *(UNIT V)*

References:

1. S. Russel and P. Norvig, *Artificial Intelligence – A Modern Approach*,3rd Edition, Pearson Education, 2010.
2. Vinod Chandra. S.S, Anand Hareendran.S., *Artificial Intelligence and Machine Learning*,PHI Publishers, 2014.
3. J. Nilsson, *Artificial Intelligence: A new Synthesis*, Elsevier Publishers, 1998.

COMPUTER GRAPHICS

(Professional Elective I)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To enable the students to learn basic and fundamental computer graphics and image synthesis techniques.
2. To expose the students to the current and emerging technologies such as OpenGL and visualize 2D and 3D objects.

UNIT I

Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and workstations 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

OpenGL: Introduction, The OpenGl API, Primitives and attributes, Color, Viewing, Control Functions, Sample Program, 3D APIs, Graphics Architecture.

Geometric objects and Transformations I: Scalars, point and Vectors, Coordinate Systems and Frames, frames in OpenGl, Translation, Rotation and Scaling.

UNIT III

Geometric objects and Transformations II: Transformations in Homogenous coordinates, Concatenation of Transformations, OpenGl Transformation Matrices.

Viewing: Classical and Computer Viewing, Positioning of the camera, Simple projections, Projections in OpenGl, Parallel projection matrices, Perspective projection matrices, projects and shadows.

UNIT IV

Lighting and Shading: Light and Matter, Light sources, the phone lighting Model, Computation of Vectors, Polygonal shading, Light sources in OpenGL, Specifications of material in OpenGl, shading of the sphere model.

UNIT V

From Vertices to Fragments: Basic implementation strategies, Four major tasks, Clipping - Line segment, Polygon, 3D, Rasterization – polygon, Bresenham's algorithm, Hidden surface removal.

Computer animation: Design of animation sequence, raster animation, computer animation languages, key frame systems, motion specifications.

Course Outcomes: At the end of the course, the student should be able to

CO 1 Know the application areas of computer graphics, overview of graphics systems and output primitives.
CO 2 Understand OpenGL as a cross platform application programming interface (API)
CO3 Apply 2D geometric transforms,2D viewing using OpenGl transformation matrices.
CO4 Create programs using OpenGL APIs for rendering 2D and 3D vector graphics.
CO 5 Use visible surface detection methods and computer animation.

Text Books:

1. Donald Hearn and M. Pauline Baker "*Computer Graphics C version*", 2nd edition, Pearson Education, 2002

2. Edward Angel, "*Interactive Computer Graphics: A Top Down Approach Using OpenGl*", 5th edition Pearson 2012.

References:

1. Marschner, Steve_ Shirley, Peter, *Fundamentals of Computer Graphics*, Fourth Edition-A K Peters, Limited, Taylor & Francis Group, 2016.
2. Steven Harrington," *Computer Graphics*" , Tata Mc Graw hill, 2008
3. Hearn, Baker & Carithers , *Computer Graphics with OpenGL*, 4th ed., Pearson New International Edition, 2013

BIG DATA ANALYTIC S AND PLATFORMS

(Professional Elective I)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand nature of big data and its platform requirements.
2. To learn software tools for big data analytics and perform map-reduce analytics using Hadoop and related tools
3. To understand HBase and Apache Spark for big data analytics

UNIT I - Introduction to Big Data and Hadoop

Classification of Digital Data, Structured Data, Semi-Structured Data, Unstructured Data, Big Data: Definition, Characteristics, Evolution, Platform Requirements, Traditional Business Intelligence versus Big Data, Introduction to Big Data Analytics

Hadoop: System Architecture, Overview of HDFS, MapReduce Model, Hadoop Evolution, Various Tools in Hadoop Ecosystem and their functional aspects.

UNIT II - HDFS and Anatomy of YARN

The design of HDFS, HDFS Concepts, The Command-Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression, serialization, Apache Avro and File-based Data structures.

Anatomy of YARN: Resource Requests, Application Lifespan, Building YARN Applications, YARN compared to MapReduce1, Scheduling in YARN, Scheduler Options, Capacity Scheduler Configuration, Fair Scheduler Configuration, Delay Scheduling, Dominant Resource Fairness

UNIT III – MapReduce and PIG

Anatomy of a MapReduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, MapReduce Types and Formats, MapReduce Features

PIG: Introduction to Pig, Execution Modes of Pig, comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions (UDFs), Data Processing Operators

UNIT IV - Hive and HBase

Hive: Introduction, Data Types and File Formats, Databases in Hive, HiveQL: Data Definition, Data Manipulation, Queries, Views, Indexes, UDF, Schema Design

HBase: Introduction, HBase v/s RDBMS, Components, Architecture, Data Models, HBase Shell, HBase Client API, Data Loading Techniques, CRUD operations

UNIT V - Introduction to Apache Spark

Apache SPARK Architecture, SPARK Applications, Jobs, Stages, and Tasks, Resilient Distributed Datasets (RDDs), Anatomy of SPARK Job Run, Executors and Cluster Managers.

SPARK SQL, Stream Data Analytics, Examples of Stream Data Analytics in SPARK, Machine learning and Advanced Analytics, SPARK-GraphX, Examples of Graph Data Analytics using SPARK-GraphX

Course Outcomes: At the end of the course, the student should be able to

- CO 1 : Identify big data problems and understand usage of various tools in Hadoop Ecosystem.
- CO 2 : Articulate working of HDFS and YARN.
- CO 3 : Write programs using MapReduce and Pig.
- CO 4 : Develop applications using Hive and HBase.
- CO 5 : Articulate Apache SPARK and its usage for Big Data Analytics.

Text Books:

1. Seema Acharya and Subhashini Chellappan*,* Big Data and Analytics, 2nd Edition, Wiley, 2019.
2. Tom White, *HADOOP: The definitive Guide*, 4th Edition, O'Reilly 2015.
3. Bill Chambers and Matei Zaharia, *Spark: The definitive Guide*, 1st Edition, O'Reilly**,** 2018.

References:

1. Shashank Tiwari, *Professional NoSQL*, WROX, 2011.
2. *Big Data Black Book*, DreamTech Publisher, 2015.

AGILE METHODOLOGY AND DEVOPS

(Professional Elective I)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. Agile Methodology gives an understanding of what Agility means, when and why to employ Agile development, the pitfalls, issues and common mistakes to watch out for, and will cover key methodologies including Scrum and Kanban.
2. Understand configuration management; continuous integration deployment, delivery and monitoring using DevOps tools such as Git, Jenkins in a practical, hands-on and interactive approach can be explored

UNIT I – Need for Agile

Establishing Need for Agile: History of the Waterfall Model,How Does Waterfall Work?,Where Is Waterfall Suitable? ,Advantages and Disadvantages of Waterfall, History of the V-Model,How Does the V-Model Work?,Where Is the V-Model Suitable?,Advantages and Disadvantages of the V-Model.
Fundamentals of Agile: A Brief History of Agile, The Agile Manifesto Core Values, Individuals and interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan, Agile Methodology Overview, Roles Within an Agile Team, Common Agile Misconceptions, Advantak ges of Agile, Disadvantages of Agile.

UNIT II – XP, Agile Scrum Framework

XP: History of Extreme Programming, Overview of Extreme Programming, Activities, Values, Principles, Practices, Rules, Extreme Programming Diagram
Agile Scrum Framework: Definition and History of Scrum, Overview of Scrum, Scrum Diagram, Scrum Roles, Scrum Ceremonies, Scrum Artifacts, Extreme Programming vs. Scrum

UNIT III – Agile Software Design and Development

Agile Software Design and Development: Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control.

UNIT IV – Devops Introduction

Introduction to DevOps: What Is DevOps? DevOps Importance and Benefits, DevOps Principles and Practices, 7'Cs of DevOps Life Cycle for Business Agility, Continuous Planning, Continuous Development, Continuous Integration, Continuous Deployment, Continuous Testing, Continuous Delivery and Monitoring, Continuous Feedback, DevOps and Continuous Testing, Steps to be followed to choose the right DevOps Tools, Selecting the Right Tools, Challenges with DevOps Implementation, Must Do Things for DevOps.

UNIT V – Tool Suits

Tool Suits: Atlassian Tools, Phabricator
Orchestration: Jenkins-Features, Example of Reference Architecture, Microsoft TFS-Features, Example of Reference Architecture, TeamCity- Features, Example of Reference Architecture, Ansible Features, Example of Reference Architecture, Bamboo- Features, Example of Reference Architecture
Source Code Management and Quality: Bitbucket- Features, Example of Reference Architecture, Crucible- Features, Example of Reference Architecture.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand the agile principles, practices.
CO 2 : Apply roles of Scrum Master and Perform Scrum Release Planning and Scrum

Sprint Planning

CO 3 : Apply Scrum with multiple, or distributed, project teams

CO 4 : Analyze Need for continuous Integration and Appreciate Continuous Deployment in Industrial Scenario

CO 5 : Implement GitHub and Jenkins for Configuration Management and Continuous Integration

Text Books:

1. Stephen Haunts, *Agile Software Development Succinctly*, Syncfusion Inc., 2015.
2. Deepak Gaikwad and Viral Thakkar, *DevOps Tools from Practitioner's Viewpoint*, Wiley, 2019.

References:

1. Robert C. Martin, *Agile Software Development- Principles, Patterns and Practices*, Prentice Hall, 2013.
2. Ken Schawber and Mike Beedle, *Agile Software Development with Scrum*, Pearson, 2001.
3. Jez Humble and David Farley, *Continuous Delivery*, Pearson Education, 2010.

MICROPROCESSORS AND CONTROLLERS

(Professional Elective I)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To appreciate pipeline implementation Architecture and Programming of 8086.
2. To interface real-world peripherals with the processor and controller
3. To design the synchronous and asynchronous communication models.

UNIT I - 8086 Architecture

8086 Architecture–Functional Diagram, Register Organization, Memory Segmentation, Signal Descriptions of 8086, Minimum and Maximum Modes, Physical Memory Organization, Timing Diagrams, Addressing Modes 0f 8086, Instruction set of 8086, Assembler Directives.

UNIT II - Memory and I/O Interfacing

SRAM Interfacing and DRAM Interfacing, 8255 PPI Architecture, Various Modes of Operation of 8255 and Interfacing with 8086, Displays, Keyboard Interfacing, Interfacing Analog to Digital converter: ADC 0808/0809, Interfacing Digital to Analog converter: DAC 0800.

UNIT III - Interrupts and Serial Communication Interface

Interrupt Structure of 8086, Vector Interrupt Table, Interrupt Service Routine, Interrupt Controller 8259 Architecture and interfacing with 8086. Introduction to DOS and BIOS Interrupts.
Serial Communication Standards, Serial Data Transfer Schemes, 8251 USART Architecture and Interfacing, RS-232.

UNIT IV - 8051 Microcontroller

Overview of 8051 Microcontroller, Architecture, I/O ports, Memory Organization, Addressing Modes and Instruction Set of 8051, Simple Programs.

UNIT V - 8051 Real Time Control

Interrupts, timer/counter and serial communication, Programming Timer Interrupts, Programming external hardware interrupts, programming the serial communication interrupts, Programming 8051 timers and counters.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Gain knowledge about pipelined processor 8086 and understand timing diagrams.
CO 2 : Interface the processor with peripheral devices.
CO 3 : Gain knowledge about interrupt structure and serial communication of 8086 microprocessor.
CO 4 : Master the 8051 architecture and programming
CO 5 : Implementing various real time controls like timers, interrupts, serial communications in 8051 micro controller

Text Books:

1. A. K. Ray and K. M. Bhurchandani, *Advanced Microprocessors and Peripherals*, 2nd Edition, Tata McGraw-Hill, 2006.
2. Kenneth J. Ayala, *The 8051 Microcontroller*, 3rd Edition, Cengage Learning, 2010.

References:

1. D.V. Hall, *Microprocessor and Interfacing*, 2ndEdition, Tata McGraw-Hill, 2006.
2. Liu and G. A. Gibson, *Micro Computer system: 8086/8088 Family Architecture, Programming and Design*, 2ndEdition, Prentice Hall, 1986.

3. Muhammad Ali Mazidi and Janice Gillispie Mazidi, *The 8051 Microcontroller and Embedded Systems*, 2nd Edition, Pearson, 2008.

INTERNET OF THINGS AND DATA SCIENCES LAB

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To offer students hands on experience in interfacing sensors with computing models.
2. To implement a complete eco system in integrating cloud offerings for IoT.
3. To develop python programs for Data Science applications.

LIST OF EXPERIMENTS

1. Start Raspberry Pi and try various Linux commands in command terminal window:
    a. ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron,
    b. chown, chgrp, ping etc.
2. Perform necessary software installation on Raspberry Pi.
3. Interface LED/Buzzer with Raspberry Pi, write a program to turn ON LED for 1 sec after every 2 seconds.
4. Interface Push button/Digital sensor (IR/LDR) with Raspberry Pi, write a program to turn ON LED when push button is pressed or at sensor detection.
5. Interface DHT11 sensor with Raspberry Pi, write a program to print temperature and humidity readings.
6. Interface motor using relay with Raspberry Pi, write a program to turn ON motor when push button is pressed.
7. Interface Bluetooth with Raspberry Pi, write a program to send sensor data
    a. to smartphone using Bluetooth.
8. Interface Bluetooth with Raspberry Pi, write a program to turn LED ON/OFF
    a. when '1'/'0' is received from smartphone using Bluetooth.
9. Interface LCD with Raspberry Pi, write a program to print temperature and
    a. humidity readings on it.
10. Write a program on Raspberry Pi to upload temperature and humidity data to cloud.
11. Write a program on Raspberry Pi to retrieve temperature and humidity data from cloud.
12. Install MySQL database on Raspberry Pi and perform basic SQL queries.
13. Write a programme in Python to predict the class of the flower based on available attributes.
14. Write a programme in Python to predict if a loan will get approved or not.
15. Write a programme in Python to predict the traffic on a new mode of transport.
16. Write a programme in Python to predict the class of user.
17. Write a programme in Python to indentify the tweets which are hate tweets and which are not.
18. Write a programme in Python to predict the age of the actors.
19. Create an Application to predict the time taken to solve a problem given the current status of the user

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Interface the sensors with Raspberry Pi using various communication models.
CO 2 : Implement real life IoT use-cases.
CO 3 : Integrate IoT with Mobile Apps.
CO 4 : Monitor the data and control devices remotely and derive a business value using cloud API.
CO 5 : Model the data and process and then create applications to solve problems using given data.

References:

1. Arshdeep Bahga and Vijay Madisetti, *Internet of Things - A Hands-on Approach,* Universities Press, 2015.
2. Peter Waher, *Learning Internet of Things,* Packt publisher, 2015.
3. Matt Richardson & Shawn Wallace, *Getting Started with Raspberry Pi,* O'Reilly (SPD), 2014.
4. Joel Grus, "*Data Science from Scratch: First Principles with Python*", O'Reilly Media
5. Kimmokarvinen and teroKarvenien, *Getting started with sensors: Measure the world with Electronics, Arduino, and Raspberry,* First Edition, Shroff/O'Reilly, 2014.
6. Richardson Matt, *Getting started with Raspberry Pi,* Shroff Publishers & Distributers Private Limited.

ADVANCED WEB TECHNOLOGIES LAB

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	30 Marks
Tutorial	:		Semester End Examination	:	70 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To develop web applications.
2. To learn the different Front-End Technologies and their usage in the development of web application development.
3. To learn the principles of server-side scripting with Node.js

List of Exeperiment:

1. Create a git repository and clone it for changes and publish the changes using git bash.
2. Working of ES6 features like fat arrows, destructuring, and function generators.
3. Create a realtime database in firebase for the student management system and explore the features of Firebase Real Time Database.
4. Explore the various node modules: os, http, etc.
5. Create a REST API and perform various CRUD operations on that.
6. Develop an express web application that can interact with a service to perform CRUD operations on student data. (Use Postman)
7. For the above application create authorized end points using JWT (JSON Web Token).
8. Create an angular application for the student management system. Include necessary pages.
9. Create a service in angular that fetches the weather information from openweathermap and the display the current and historical weather information using graphical representation using chart.js
10. Develop an Angular application to perform CRUD operation on a Web service using HTTP.
11. Bundle the angular application created under week 9 and deploy it to the github and host it as a github pages.
12. Develop an react application for student management system. Include necessary pages.
13. Access a web service in react that fetches the weather information from openweathermap and the display the current and historical weather information using graphical representation using chart.js.
14. Bundle the react application created under week 13 and deploy it to the github and host it as a github pages.
15. Create a TODO application in react with necessary components and deploy it into github.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : To understand the use of prototyping, ES6 classes, advanced JavaScript concepts and make use of the XMLHttpRequest API.
CO 2 : To define, compare and use of the four types of NoSQL Databases.
CO 3 : To design and build robust REST APIs using Node.js, demonstrate the Express framework and work with Typescript.
CO 4 : To implement one way / two way data binding for data interpolation, dependency injection and design single page application using Angular2.
CO 5 : To create smaller components to build Interactive User interfaces using ReactJS.

References: