**B.B.A.(C.A.) Semester –III**

**Course Code: CA-301**

**Subject: Digital Marketing**

**Objectives:**

1. The aim of this syllabus is to give knowledge about using digital marketing in and as business.

2. To make SWOT analysis, SEO optimization and use of various digital marketing tools.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of**  **Lectures** |
|  | **E-Commerce**   * 1. Introduction   2. Understanding Internet Marketing   3. Search Engine Optimization   4. Search Engine Marketing   5. Email Marketing   6. Digital Display Marketing | **4** |
|  | **Introduction to New Age Media (Digital) Marketing**   * 1. What is Digital Marketing   2. Digital vs. Real Marketing   3. Digital Marketing Channels   4. Types of Digital Marketing(Overview)-Internet Marketing ,Social Media Marketing, Mobile Marketing | **4** |
|  | **Creating Initial Digital Marketing Plan**   * 1. Content management   2. SWOT analysis:  Strengths, Weaknesses, Opportunities, and Threats   3. Target group analysis   EXERCISE: Define a target group | **4** |
|  | **Marketing using Web Sites**   * 1. Web design   2. Optimization of Web sites   3. MS Expression Web   EXERCISE: Creating web sites, MS Expression | **4** |
|  | **Search Engine Optimization**   * 1. SEO Optimization   2. Writing the SEO content   EXERCISE: Writing the SEO content | **4** |
|  | **Customer Relationship Management**   * 1. Introduction to CRM   2. CRM platform   3. CRM models   EXERCISE: CRM strategy | **4** |
|  | **Social Media Marketing**   * 1. Understanding Social Media Marketing   2. Social Networking (Facebook, Linkedin, Twitter, etc.)   Social Media (Blogging, Video Sharing - Youtube,  Photosharing – Instagram, Podcasts)   * 1. Web analytics - levels   2. Modes of Social Media Marketing-      1. **Creating a Facebook page** Visual identity of a Facebook page , Types of publications, Facebook Ads , Creating Facebook Ads , Ads Visibility      2. **Business opportunities and Instagram options** Optimization of Instagram profiles , Integrating Instagram with a Web Site and other social networks ,Keeping up with posts      3. **Business tools on LinkedIn** Creating campaigns on LinkedIn , Analyzing visitation on LinkedIn      4. **Creating business accounts on YouTube** YouTube ,Advertising , YouTube Analytics      5. **E-mail marketing** E-mail marketing plan , E-mail marketing campaign analysis , Keeping up with conversions   3. Digital Marketing tools: Google Ads, FaceBook   Ads, Google Analytic, Zapier, Google Keyword Planner  EXERCISE: Social Media Marketing plan.  EXERCISE: Making a Facebook page and Google Ads | **1**  **2**  **2**  **3**  **3**  **3**  **3**  **3**  **(20)** |
|  | **Digital Marketing Budgeting**   * 1. Resource planning   2. Cost estimating   3. Cost budgeting   4. Cost control | **4** |
| **Total** | | **48** |

# Reference Books:

1. Digital Marketing for Dummies By Ryan Deiss and Russ Hennesberry
2. Advertising and Promotion: An Integrated Marketing Communications Perspective,

George Belch, San Diego University Michael Belch, San Diego University

1. Advertising Management: Rajeev Batra, John G. Myers, David A. Aaker
2. Belch: Advertising & Promotions (TMH)
3. The Social Media Bible: Tactics, Tools, & Strategies for Business Success by Lon Safko
4. Web Analytics 2.0 – Avinash Kaushik

**B.B.A(C.A) Semester – III**

**Course Code: CA-302**

**Subject : Data Structure**

**Objectives:**

1. To understand the concepts of ADTs
2. To learn linear data structures – lists, stacks, and queues
3. To understand sorting, searching and hashing algorithms
4. To apply Tree and Graph structures

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of Lectures** |
| **1** | **Basic Concept and Introduction to Data Structure**  1.1 Pointers and dynamic memory allocation  1.2 Algorithm-Definition and characteristics  1.3 Algorithm Analysis -Space Complexity -Time Complexity -Asymptotic Notation Introduction to Data structure  1.4 Types of Data structure  1.5 Abstract Data Types (ADT) Introduction to Arrays and Structure  1.6 Types of array and Representation of array  1.7 Polynomial - Polynomial Representation - Evaluation of Polynomial - Addition of Polynomial  1.8 Self Referential Structure | **5** |
| **2** | **Linear data structures**  2.1 Introduction to Arrays - array representation  2.2 Sorting algorithms with efficiency  - Bubble sort, Insertion sort, Merge sort, Quick Sort, Selection Sort  2.3 Searching techniques –Linear Search, Binary search | **6** |
| **3** | **Linked List**  3.1 Introduction to Linked List  3.2 Implementation of Linked List – Static & Dynamic representation,  3.3 Types of Linked List  - Singly Linked list(All type of operation)  - Doubly Linked list (Create , Display)  - Circularly Singly Linked list (Create, Display)  - Circularly Doubly Linked list (Create, Display)  3.4 Generalized linked list – Concept and Representation | **6** |
| **4** | **Stacks**  4.1 Introduction  4.2 Representation- Static & Dynamic  4.3 Primitive Operations on stack  4.4 Application of Stack  4.5 Conversion of Infix, prefix, postfix , Evaluation of postfix and prefix  4.6 Simulating recursion using stack | **8** |
| **5** | **Queues**  5.1 Introduction  5.2 Representation - Static & Dynamic  5.3 Primitive Operations on Queue  5.4 Circular queue, priority queue  5.5 Concept of doubly ended queue | **4** |
| **6** | **Trees**  6.1 Concept & Terminologies  6.2 Binary tree, binary search tree  6.3 Representation – Static and Dynamic  6.4 Operations on BT and BST – create, Insert, delete, , counting leaf, non-leaf & total nodes ,  6.5 Tree Traversals (preorder, inorder, postorder)  6.6 Application - Heap sort  6.7 Height balanced tree- AVL trees- Rotations, AVL tree examples. | **12** |
| **7** | **Graph**  7.1 Concept & terminologies  7.2 Graph Representation – Adjacency matrix, adjacency list, inverse Adjacency list, adjacency multilist, orthogonal list  7.3 Degree of Graph  7.4 Traversals – BFS and DFS  7.5 Applications – AOV network – topological sort, AOE network – critical Path | **7** |
| **Total** | | **48** |

**Reference Books:**

1. Fundamentals of Data Structures ---- By Horowitz Sahani (Galgotia)

2. Data Structures using C and C++ --- By YedidyahLangsam, Aaron M.

Tenenbaum, Moshe J. Augenstein

3. Introduction to Data Structures using C---By Ashok Kamthane

4. Data Structures using C --- Bandopadhyay&Dey (Pearson)

5. Data Structures using C ---By Srivastava BPB Publication.

**B.B.A. (C.A.) Semester –III**

**Course Code: CA-303**

**Subject: Software Engineering**

**Objectives:**

1. To understand System concepts.

2. To understand Software Engineering concepts.

3. To understand the applications of Software Engineering concepts and Design in Software development

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of lectures** |
| **1** | **Introduction to System Concepts**  1.1 Definition  1.2 Basic Components  1.3 Elements of the System  1.4 Types of System  1.5 System Characteristics | **4** |
| **2** | **Introduction to Software Engineering**  2.1 Definition of Software  2.2 Characteristics of Software  2.3 Definition of Software Engineering  2.4 Need for Software Engineering  2.5 Mc Call’s Quality factors  2.6 The Software Process  2.7 Software Product and Process  2.8 V& V Model | **6** |
| **3** | **Software Development Life Cycle**  3.1 Introduction  3.2 Activities of SDLC  3.3 A Generic Process Model  3.4 SDLC  3.5 Waterfall Model  3.6 Incremental Process Models  3.7 Prototyping Model  3.8 Spiral Model | **8** |
| **4** | **Requirement Engineering**  4.1 Introduction  4.2 Requirement Elicitation  4.3 Requirement Elaboration  4.4 Requirement Gathering  4.5 Feasibility study  4.6 Fact Finding Techniques  4.7 SRS Format | **8** |
| **5** | **Analysis And Design Tools**  5.1 Decision Tree and Decision Table  5.2 Data Flow Diagrams (DFD) (Up to 2nd level)  5.3 Data Dictionary  5.4 Elements of DD  5.5 Advantages and Disadvantages of DD  5.6 Input and Output Design  5.7 Structured Design Concepts  5.8 Structure Chart  5.9 Coupling and Cohesion  5.10 Compulsory Case Studies on above topics | **12** |
| **6** | **Software Testing**  6.1 Definition  6.2 Software testing Process  6.3 Unit Testing  6.4 Integration Testing  6.5 System Testing | **6** |
| **7** | **Software Maintenance and Software Re-Engineering**  7.1 Maintenance definition and types  7.2 Software reengineering  7.3 Reverse Engineering  7.4 Restructuring and forward Engineering. | **4** |
| **Total** | | **48** |

**Reference Books:**

1. Software Engineering: A Practitioner’s Approach- Roger S. Pressman, McGraw hill International Editions 2010(Seventh Edition)

2. System Analysis, Design and Introduction to Software Engineering (SADSE) - S. Parthsarthy, B.W. Khalkar

3. Analysis and Design of Information Systems(Second Edition) - James A. Senn, McGraw Hill

4. System Analysis and Design- Elias Awad, Galgotia Publication, Second Edition

**B.B.A.(C.A.) Semester – III**

**Course Code: CA- 304**

**Subject: Angular - JS**

**Objectives:**

* By the end of this course, the students should be able to Understand Client Side MVC and SPA
* Explore AngularJS Component
* Develop an AngularJS Single Page Application
* Create and bind controllers with Javascript
* Apply filter in AngularJS application

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topics** | **No. of Lectures** |
| **1** | **AngularJS Core Concepts:**   * 1. What is AngularJS?   2. Difference between Javasript and Angular JS   3. Advantages of Angular   4. AngularJS MVC Architecture   5. Introduction to SPA   6. Setting up the environment   7. First App using MVC architecture | **8** |
| **2** | **AngularJS Directives and Expressions:**  2.1 Understanding ng attributes  ng-app, ng-init, ng-model, ng-controller, ng-bind,  ng-repeat, ng-show, ng-readonly, ng-disabled,  ng-if, ng-click  2.2 Expression and Data Binding  2.3 Working with directives | **10** |
| **3** | **AngularJS Modules, Controller, View and Scope:**  3.1 Angular Modules  3.2 Angular Controller  3.3 Angular View  3.4 Scope hierarchy | **10** |
| **4** | **Filter, Forms and Ajax Filters**  4.1 Built-in filters  - upper case and lower case filters, date ,currency and  number formatting ,orderBy, filter ,custom filter,  4.2 Angular JS Forms  – Working with AngularJS forms, model binding,  form controller ,Using CSS classes, form events ,  custom model update triggers ,custom validation,  $http service ,  4.3 Ajax implementation using $http | **12** |
| **5** | **Dependency Injection, Services**  5.1 What is dependency injection?  5.2 Understanding services  5.3 Using built-in service  5.4 Creating custom service,  5.5 Injecting dependency in service | **8** |
| **Total** | | **48** |

**Reference Books:**

1. Beginning Angular with Typescript (updated to Angular 5) by  [Greg Lim](https://www.goodreads.com/author/show/16410277.Greg_Lim)

# Mastering Web Application Development with AngularJS by  [Pawel Kozlowski](https://www.goodreads.com/author/show/7132207.Pawel_Kozlowski), [Peter Bacon Darwin](https://www.goodreads.com/author/show/7132206.Peter_Bacon_Darwin)

1. <https://www.tutorialsteacher.com/angularjs/angularjs-scope>

**B.B.A.(C.A.) Semester – III**

**Course Code: CA-304**

**Subject: Python**

**Objectives:**

1. Understand importance of Python programming.
2. Learn various Built-in data structure and functions used in python
3. Write basic and general-purpose programs within few lines of code;

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of lectures** |
| **1** | **Basic Python**   * 1. Python identifiers and reserved words   2. Lines and indentation, multi-line statements Comments   3. Input/output with print and input functions   4. Command line arguments and processing command line arguments   5. Standard data types - basic, none, Boolean (true & False), numbers   6. Data type conversion | **4** |
| **2** | **Operators, Loop and Control Statements in Python**   * 1. Python basic operators (Arithmetic, comparison, assignment, bitwise logical)   2. Python membership operators (in & not in)   3. Python identity operators (is & is not)   4. Operator precedence   5. Control Statements, Python loops, Iterating by subsequence index   6. Loop control statements (break, continue, pass)   7. Mathematical functions and constants (import math), Random number functions | **6** |
| **3** | **Python strings**   * 1. Concept and escape characters   2. String special operations   3. String formatting operator   4. Single quotes, Double quotes, Triple quotes   5. Raw String, Unicode strings   6. Built-in String methods.   7. Python Lists - concept, creating and accessing elements, updating & deleting lists, basic list operations, reverse Indexing, slicing and Matrices   8. built-in List functions   9. Functional programming tools - filter(), map(), and reduce()   10. Using Lists as stacks and Queues, List comprehensions | **6** |
| **4** | **Python tuples and sets**   * 1. Creating & deleting tuples   2. Accessing values in a tuple   3. Updating tuples, delete tuple elements   4. Basic tuple operations   5. Indexing, slicing and Matrices, built- in tuple functions.   6. Sets - Concept, operations. | **4** |
| **5** | **Python Dictionary**   * 1. Concept (mutable)   2. Creating and accessing values in a dictionary   3. Updating dictionary, delete dictionary elements   4. Properties of dictionary keys   5. built-in dictionary functions and methods. | **4** |
| **6** | **Functions**   * 1. Defining and calling a function   2. Function arguments - Pass by value, Keyword Arguments, default arguments   3. Scope of variable - basic rules   4. Call by Reference   5. Order of arguments (positional, extra & keyword)   6. Anonymous functions   7. Recursion   8. Treatment of Input and Output Arguments   9. Unpacking argument lists   10. Lambda forms   11. Function Objects   12. Function ducktyping & polymorphism   13. Generators (functions and expressions) and iterators | **7** |
| **7** | **Files and Directories**   * 1. Creating files   2. Operations on files (open, close, read, write)   3. File object attributes, file positions, Listing Files in a Directory   4. Testing File Types  Removing Files and Directories   5. Copying and Renaming Files   6. Splitting Pathnames   7. Creating and Moving to Directories   8. Traversing Directory Trees   9. Illustrative programs: word count, copy file | **6** |
| **8** | **Python Classes / Objects**   * 1. Object oriented programming and classes in Python - creating classes, instance objects, accessing members   2. Data hiding (the double underscore prefix)   3. Built-in class attributes   4. Garbage collection : the constructor   5. Overloading methods and operators   6. Inheritance - implementing a subclass, overriding methods   7. Recursive calls to methods   8. Class variables, class methods, and static methods | **7** |
| **9** | **Python Exceptions**   * 1. Exception handling: assert statement   2. Except clause - with no exceptions and multiple exceptions   3. Try - finally, raising exceptions, user-defined exceptions | **4** |
| **Total** | | **48** |

# Reference Books:

1. Python Programming : An Introduction to Computer Science

by John Zelle (Author), Michael Smith (Author)

1. Python Essential Reference 4th Edition - by David Beazley (Author)
2. Python: The Complete Reference Paperback – 20 Mar 2018 by Martin C. Brown (Author)

**B.B.A.(C.A.) Semester – III**

**Course Code: CA- 305**

**Subject: Big Data**

**Objectives:**

1. To enable learners to develop expert knowledge and analytical skills in current and developing areas of analysis statistics, and machine learning

2. To enable the learner to identify, develop and apply detailed analytical, creative, problem solving skills.

3. Provide the learner with a comprehensive platform for career development, innovation and further study.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of lectures** |
| **1** | **INTRODUCTION TO BIG DATA**  1.1 Introduction to Big Data  1.2 Types of Digital Data  1.3 Big Data Analytics  1.4 Application of Big data | **5** |
| **2** | **Introduction to Data Science**  2.1 Basics of Data Analytics  2.2 Types of Analytics –  2.2.1 Descriptive,  2.2.2 Predictive,  2.2.3 Prescriptive  2.2.4 Statistical Inference  2.3 Populations and samples  2.3.1 Statistical modelling,  2.3.2 Probability  2.3.3 Distribution  2.3.4 Correlation  2.3.5 Regression | **10** |
| **3** | **Introduction to Machine Learning**   * 1. Basics of Machine Leaning   2. Supervised Machine Learning   3.2.1 K- Nearest-Neighbours,  3.2.2 Naïve Bayes  3.2.3 Decision tree  3.2.4 Support Vector Machines  3.3 Unsupervised Machine Learning  3.3.1 Cluster analysis  3.3.2 K means  3.3.3 EM Algorithm  3.3.4 Association Rule Mining   * + 1. Apriori algorithms   3.4 Regression Analysis  3.4.1 Linear Regression  3.4.2 Nonlinear Regression | **20** |
| **4** | **Data Analytics with R/ Weka Machine Learning**  4.1 Introduction  4.2 Data Manipulation  4.3 Data Visualization  4.4 Data Analysis | **13** |
| **Total** | | **48** |

# Reference Books:

1. Seema Acharya, SubhasiniChellappan, "Big Data Analytics" Wiley 2015.
2. Jay Liebowitz, “Big Data and Business Analytics” Auerbach Publications, CRC press (2013)
3. ArvindSathi, “BigDataAnalytics: Disruptive Technologies for Changing the Game”, MC Press, 2012

**B.B.A.(C.A.) Semester – III**

**Course Code: CA-305**

**Subject: BlockChain**

**PREREQUISITES:**

This course is highly technical in nature and would require the student to be comfortable with coding. To prepare for the class all students MUST:

* Understanding of basic programming language like Java, or Javascript.
* Understanding of PKI and Docker.

**WHAT YOU’LL LEARN**

* Understand what and why of Blockchain
* Explore the major components of Blockchain
* Learn about Bitcoin, Cryptocurrency, Ethereum
* Deploy and exercise example smart contracts
* Identify a use case for a Blockchain application
* Create your own Blockchain network application

**COURSE OBJECTIVES**

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

1. Understand how blockchain systems (mainly Bitcoin and Ethereum) work,
2. To securely interact with them,
3. Design, build, and deploy smart contracts and distributed applications,
4. Integrate ideas from blockchain technology into their own projects.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of**  **Lectures** |
| **1** | **Introduction To Blockchain**   * 1. Digital Trust   2. Asset   3. Transactions   4. Distributed Ledger Technology   5. Types of network   6. Components of blockchain or DLT   7. Ledger   1.7.1. Blocks  1.7.2. Blockchain  1.8 PKI and Cryptography  1.8.1. Private keys  1.8.2. Public keys  1.8.3. Hashing  1.8.4. Digital Signature  1.9. Consensus  1.9.1. Byzantine Fault  1.9.2. Proof of Work  1.9.3. Poof of Stake  1.10. Security  1.10.1.DDos  1.11 Cryptocurrency  1.12.Digital Token | **12** |
| **2.** | **How Blockchain Works**  2.1 How Blockchain Works  2.2. Structure of Blockchain  2.3.Block  2.4. Hash  2.5. Blockchain  2.6. Distributed  2.7. Lifecycle of Blockchain  2.8. Smart Contract  2.9. Consensus Algorithm  2.10 Proof of Work  2.11 Proof of Stake  2.12 Practical Byzantine  2.13 Fault Tolerance  2.14 Actors of Blockchain  2.15 Blockchain developer  2.16 Blockchain operator  2.17 Blockchain regulator  2.18 Blockchain user  2.19 Membership service provider  2.20 Building A Small Blockchain Application | **12** |
| **3.** | **Introduction to Bitcoin**  3.1 Currency  3.2 Double Spending  3.3 Cryptocurrency  3.4 P2P Payment Gateway  3.5 Wallet  3.6 Mining | **8** |
| **4.** | **Ethereum**   * 1. Ethereum network   2. EVM   3. Transaction fee   4. Mist   5. Ether, gas   6. Solidity - Smart contracts   7. Truffle   8. Web3   9. Design and issue Cryptocurrency   10. Mining   11. DApps   12. DAO | **8** |
| **5** | **Introduction To Hyperledger Fabric V1.1**  5.1. Introduction to Hyperledger  5.2 What is Hyperledger  5.3 Why Hyperledger  5.4 Where can Hyperledger be used  5.5 Hyperledger Architecture  5.6 Membership  5.7 Blockchain  5.8 Transaction  5.9 Chaincode  5.10 Hyperledger Fabric  5.11 Features of Hyperledger | **8** |
| **Total** | | **48** |

**References:**

**Text Book**

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder,

Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton

University Press (July 19, 2016).

**Reference Books**

1. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies

2. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System

3. DR. Gavin Wood, “ETHEREUM: A Secure Decentralized Transaction Ledger,”Yellow paper.2014.

4. Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli, A survey of attacks on Ethereum smart contracts

**B.B.A.(C.A.) Semester –IV**

**Course Code: CA-401**

**Subject: Networking**

**Objectives:**

1. To gain knowledge about Computer Networks concepts.

2. To know about working of networking models, addresses, transmission medias and

connectivity devices.

3. To acquire information about network security and cryptography.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of Lectures** |
| **1** | **Introduction to Computer Network**  1.1Basics of Computer Network  1.1.1Definition  1.1.2Goals  1.1.3Applications,  1.1.4Network Hardware –Broadcast, Point to Point  1.1.5Components of Data Communication   * 1. Network Topologies   1.2.1Mesh  1.2.2 Star,  1.2.3 Bus,  1.2.4Ring  1.3Types of Networks  1.3.1LAN,MAN,WAN,  1.3.2 Internetwork,   * + 1. Wireless Network   1.4 Modes of Communication  1.4.1 Simplex,  1.4.2 Half Duplex,  1.4.3 Full Duplex  1.5. Server Based LANs & Peer-to-Peer LANs  1.6. Protocols and Standards  1.7. Network Software  1.7.1 Protocol Hierarchies,Layers, Peers,Interfaces  1.7.2 Design Issues of the Layers  1.7.3 Connection Oriented and Connectionless Service | **10** |
| **2** | **Network Models**  2.1 OSI Reference Model : Functions of each Layer  2.2 TCP/IP Reference Model, Comparison of OSI and TCP/IP Reference Model  2.3 TCP/IP Protocol Suite   * 1. Addressing   2.4.1Physical Addresses  2.4.2 Logical Addresses  2.4.3Port Addresses,  2.4.4 Specific Addresses   * 1. IP Addressing   2.5.1 ClassfullAddressing  2.5.2 Classless Addressing | **8** |
| **3** | **Transmission Media**  3.1 Introduction, Types of Transmission Media   * 1. Guided Media:   3.2.1Twisted Pair Cable- Physical Structure, Categories, Connectors &Applications  3.2.2Coaxial Cable – Physical Structure, Standards, Connectors & Applications  3.2.3Fiber Optic Cable- Physical Structure, Propagation Modes, Connectors & Applications   * 1. Unguided Media:   3.3.1Electromagnetic Spectrum for Wireless Communication  3.3.2Propagation Modes Ground, Sky, Line-of-Sight  3.3.3Wireless Transmission: Radio Waves, Microwaves, Infrared | **8** |
| **4** | **Wired and Wireless LAN**  4.1 IEEE Standards  4.2 Standard Ethernet MAC Sublayer, Physical Layer  4.3 Fast Ethernet – Goals, MAC Sublayer, Topology, Implementation  4.4 Gigabit Ethernet – Goals, MAC Sublayer, Topology, Implementation  4.5 Ten-Gigabit Ethernet – Goals, MAC Sublayer, Physical Layer  4.6 Backbone Networks -Bus Backbone, Star Backbone  4.7 Virtual LANs Membership, IEEE standards advantages   * 1. Wireless LAN   4.8.1 IEEE 802.11 Architecture,  4.8.2 Bluetooth Architecture (Piconet, Scatternet) | **8** |
| **5** | **Network Devices**   * 1. Network Connectivity Devices   5.1.1 Active and Passive Hubs  5.1.2 Repeaters  5.1.3 Bridges- Types of Bridges  5.1.4 Switches  5.1.5 Router  5.1.6 Gateways | **6** |
| **6** | **Network Security**  6.1 Introduction  6.2 Need for Security   * 1. Security Services :   6.3.1 Message- -Confidentiality, Integrity, Authentication, Non repudiation.  6.3.2 Entity (User)- Authentication.  6.4 Types of Attack  6.5 Cryptography, PlainText,Cipher Text, Encryption,Decryption, Symmetric Key and Asymmetric Key Cryptography  6.6 SubstitutionTechniques, Caesar Cipher,and Transposition Cipher (Problems should be covered.)  6.7 Firewalls- Packet Filter firewall, Proxy firewall  6.8 Steganography,Copyright | **8** |
| **Total** | | **48** |

**Reference Books:**

1. Computer Networks by Andrew Tanenbaum, Pearson Education.[4th Edition]

2. Data Communication and Networking by BehrouzForouzan, TATA McGraw Hill. .[4th Edition]

**B.B.A.(C.A.) Semester –IV**

**Course Code: CA-402**

**Subject: Object Oriented Concepts Through CPP**

**Objectives:**

1. Acquire an understanding of basic object-oriented concepts and the issues involved in effective class design.
2. Enable students to write programs using C++ features like operator overloading, constructor and destructor, inheritance, polymorphism and exception handling.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of Lectures** |
| **1** | **Introduction to C++**   * 1. Basic concepts, features, advantages and applications of OOP   2. Introduction, applications and features of C++   3. Input and Output operator in C++   4. Simple C++ program | **2** |
| **2** | **Beginning with C++**  2.1 Data type and Keywords  2.2 Declaration of variables, dynamic initialization of variables, reference variable  2.3 Operators:  2.3.1 Scope resolution operator  2.3.2 Memory management operators  2.4 Manipulators  2.5 Functions:  2.5.1 Function prototyping, call by reference and return by reference  2.5.2 Inline functions  2.6 Default arguments | **6** |
| **3** | **Classes and Objects**  3.1 Structure and class, Class, Object  3.2 Access specifiers, defining data member  3.3 Defining member functions inside and outside class definition.  3.4 Simple C++ program using class  3.5 Memory allocation for objects  3.6 Static data members and static member functions  3.7 Array of objects, objects as a function argument  3.8 Friend function and Friend class  3.9 Function returning objects | **8** |
| **4** | **Constructors and Destructors**  4.1 Constructors  4.2 Types of constructor : Default, Parameterized, Copy  4.3 Multiple constructors in a class  4.4 Constructors with default argument  4.5 Dynamic initialization of constructor  4.6 Dynamic constructor  4.7 Destructor | **6** |
| **6** | **Inheritance**  6.1 Introduction  6.2 Defining Base class and Derived class  6.3 Types of Inheritance  6.4 Virtual Base Class  6.5 Abstract class  6.6 Constructors in derived class | **6** |
| **7** | **Polymorphism**   * 1. Compile Time Polymorphism      1. Introduction, rules for overloading operators      2. Function overloading      3. Operator Overloading unary and binary      4. Operator Overloading using friend function      5. Overloading insertion and extraction operators      6. String manipulation using operator overloading   2. Runtime Polymorphism      1. this Pointer, pointers to objects, pointer to derived classes      2. Virtual functions and pure virtual functions | **8** |
| **8** | **Managing console I/O operations**   * 1. C++ streams and C++ stream classes   2. Unformatted I/O operations   3. Formatted console I/O operations   4. Output formatting using manipulators   5. User defined manipulators | **3** |
| **9** | **Working with Files**  9.1 Stream Classes for File operations  9.2 File operations - Opening, Closing and updating  9.3 File updating with random access.  9.4 Error handling during File operations  9.5 Command Line arguments | **6** |
| **10** | **Templates**   * 1. 10.1 Introduction   10.2 Class Template and class template with multiple parameters   * 1. 10.3 Function Template and function template with multiple parameter   10.4 Exception Handling Introduction | **3** |
| **Total** | | **48** |

# Reference Books:

* 1. Object Oriented programming with C++ by E Balagurusamy
  2. Object Oriented Programming with C++ by Robert Lafore
  3. The Complete Reference C++ by Herbert Schildt

**B.B.A. (C.A.) Semester-IV**

**Subject**: **Operating System**

**Course Code: CA-403**

**Objectives:**

1. To know the services provided by Operating System
2. To know the scheduling concept
3. To understand design issues related to memory management and various related algorithms.
4. To understand design issues related to File management and various related algorithms

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of Lectures** |
| **1** | **Introduction to Operating System**  1.1 What is operating system  1.2 Computer system architecture  1.3 Services provided by OS  1.4 Types of OS  1.5 Operating System Structure –  - Simple structure  -Layered approach  -Micro kernels  -Modules  1.6 Virtual Machines – Introduction, Benefits | **3** |
| **2** | **System Structure**  2.1 User operating system Interface  2.2 System Calls –  - Process or job control  - Device Management  - File Management  2.3 System Program  2.4 Operating System Structure | **3** |
| **3** | **Process Management**  3.1 Process Concept –  - The process  - Process states  - Process control block  3.2 Process Scheduling –  - Scheduling queues  - Schedulers  - Context Switch  3.3 Operation on Process –  - Process Creation  -Process Termination  3.4 Interprocess Communication –  - Shared memory system  - Message passing systems. | **4** |
| **4** | **CPU Scheduling**  4.1 What is scheduling  4.2 Scheduling Concepts –  - CPU- I/O Burst Cycle  - CPU Scheduler  - Preemptive and Non-preemptive scheduling  - Dispatcher  4.3 Scheduling criteria  4.4 Scheduling Algorithms –  - FCFS  - SJF ( Preemptive& non-preemptive)  - Priority Scheduling (Preemptive& Non- preemptive)  - Round Robin Scheduling  - Multilevel Queues  - Multilevel Feedback queues | **6** |
| **5** | **Process Synchronization**  5.1 Introduction  5.2 Critical section problem  5.3 Semaphores –  - Concept  - Implementation  - Deadlock & Starvation  - Types of Semaphores  5.4 Classical Problems of synchronization –  -Bounded buffer problem  - Readers & writers problem  - Dining Philosophers problem | **6** |
| **6** | **Deadlock**  6.1 Introduction  6.2 Deadlock Characterization  6.3 Necessary Condition  6.4 Deadlock Handling Technique –  - Deadlock Prevention  - Deadlock Avoidance –  - Safe State  - Resource allocation graph algorithm  - Bankers algorithm  - Deadlock Detection  - Recovery from Deadlock –  -Process Termination  -Resource Preemption | **7** |
| **7** | **Memory Management**  7.1.Background –  -Basic hardware  - Address binding  - Logical versus physical address space  - Dynamic loading  - Dynamic linking and shared libraries  7.2 Swapping  7.3 Contiguous Memory Allocation –  - Memory mapping and protection  -Memory allocation  - Fragmentation  7.4 Paging –  - Basic Method  - Hardware support  - Protection  - Shared Pages  7.5 Segmentation –  - Basic concept  - Hardware  7.6 Virtual Memory Management –  - Background  - Demand paging  - Performance of demand paging  - Page replacement –  - FIFO  - OPT  - LRU  - Second chance page replacement  - MFU  - LFU | **8** |
| **8** | **File System**  8.1 Introduction & File concepts (file attributes, Operations on files)  8.2 Access methods –  - Sequential access  - Direct access  8.3 File structure –  - Allocation methods  - Contiguous allocation  - Linked Allocation  - Indexed Allocation  8.4 Free Space Management –  - Bit Vector  - Linked List  - Grouping  - Counting | **7** |
| **9** | **I/O System**  9.1 Introduction  9.2 I/O Hardware  9.3 Application of I/O Interface  9.4 Kernel I/O Subsystem  9.5 Disk Scheduling –  - FCFS  - Shortest Seek time first  - SCAN  - C- SCAN  - C- Look | **4** |
| **Total** | | **48** |

**Reference Books:**

1. Operating System Concepts - Siberchatz, Galvin, Gagne (8th Edition).

2. Operating Systems : Principles and Design – Pabitra Pal Choudhary (PHI Learning Private Limited)

**B.B.A.(C.A.) Semester – IV**

**Course Code: CA- 404**

**Subject: Node - JS**

**Objectives:**

1. Understand the JavaScript and technical concepts behind Node JS
2. Structure a Node application in modules
3. Understand and use the Event Emitter
4. Understand Buffers, Streams, and Pipes
5. Build a Web Server in Node and understand how it really works
6. Connect to a SQL or Mongo database in Node

**Pre-requisite / Target Audience:**

1. Basic Knowledge of JavaScript and OOPS
2. Knowledge in async programming will be added advantage

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topics** | **No. of**  **Lectures** |
| **1** | **Introduction to Node JS**   * 1. Introduction   2. What is Node JS?   3. Advantages of Node JS   4. Traditional Web Server Model   5. Node.js Process Model   6. Install Node.js on Windows   7. Working in REPL | **8** |
| **2** | **Node JS Modules**  2.1Functions  2.2 Buffer  2.3 Module  2.4 Module Types  2.5 Core Modules  2.6 Local Modules  2.7 Module. Exports | **10** |
| **3** | **Node Package Manager**  3.1 What is NPM ?  3.2 Installing Packages Locally  3.3 Adding dependency in package.json  3.4 Installing packages globally  3.5 Updating packages | **6** |
| **4** | **Web server**  4.1 Creating web server  4.2 Handling http requests  4.3 Sending requests | **6** |
| **5** | **File System**  5.1 Fs.readFile  5.2 Writing a File  5.3 Writing a file asynchronously  5.4 Opening a file  5.5 Deleting a file  5.6 Other IO Operations | **8** |
| **6** | **Events**  6.1 EventEmitter class  6.2 Returning event emitter  6.3 Inhering events | **4** |
| **7** | **Database connectivity**  7.1 Connection string  7.2 Configuring  7.3 Working with select command  7.4 Updating records  7.5 Deleting records | **6** |
| **Total** | | **48** |

**Reference Books:**

1. Node.js complete reference guid , velentin Bojinov, David Herron, Dioge Resende, packt Publishing ltd
2. Mastering Nod.js By Sandro Pasquali , packt Publishing
3. Smashing Node.js Javascript Everywhere , Guillermo Rauch, John wiley & Sons

**B.B.A.(C.A.) Semester – IV**

**Course Code: CA- 404**

**Subject: PHP**

**Objectives:**

1. Understand how server-side programming works on the web.
2. Using PHP built-in functions and creating custom functions
3. Understanding POST and GET in form submission.
4. How to receive and process form submission data.
5. Read and process data in a MySQL database.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of Lectures** |
| **1** | **PHP Basics**   * 1. Setting up a development environment   2. Variables, numbers and strings   3. Calculations with PHP   4. Using Arrays | **6** |
| **2** | **Control Structures and Loops**   * 1. Conditional Statements   2. Using Loops for Repetitive tasks   3. Combing Loops and Arrays | **7** |
| **3** | **Functions, Objects and Errors**   * 1. PHP's Built-in functions   2. Creating Custom functions   3. Passing Values by Reference   4. Understanding Objects | **7** |
| **4** | **Working with Forms**   * 1. Building a Form   2. Processing a Form's Data   3. Differences between POST and GET   4. Preserving User Input | **7** |
| **5** | **More with Forms**   * 1. Dealing with checkboxes and radiobuttons   2. Retrieving values from lists   3. Validating and restricting data   4. Sending Email | **7** |
| **6** | **Storing and Protecting Data**   * 1. Setting and Reading Cookies   2. Protecting Online Files   3. Understanding Session Variables | **7** |
| **7** | **MySQL Database Overview**   * 1. phpMyAdmin Overview   2. Using a MySQL Database   3. Reading and Writing Data | **7** |
| **Total** | | **48** |

# Reference Books:

1. Php: A Beginner's Guide 1st EditionMcGraw-Hill Osborne Media; 1 edition by Vikram Vaswani
2. Murach’s PHP and MySQL (2nd Edition)by Joel Murach and Ray Harris
3. PHP: The Complete Reference Paperback – 1 Jul 2017by Steven Holzner (Author)

**B.B.A.(C.A.) Semester – IV**

**Course Code: CA- 405**

**Subject: Internet of Things**

**Pre-Requisite: Basic understanding of electronics and microprocessors.**

**Course Objectives:**

1. The Internet of Things (IoT) is aimed at enabling the interconnection and integration of

the physical world and the cyber space.

2. To learn about SoC architectures, programming Raspberry Pi and implementation of

Internet of things and protocols.

**Expected Learning Outcomes:**

1. Enable learners to understand System On Chip Architectures.

2. Introduction and preparing Raspberry Pi with hardware and installation.

3. Learn physical interfaces and electronics of Raspberry Pi and program them using practical

4. Learn how to design IoT based prototypes.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Topic** | **No. of**  **Lectures** |
| **1** | **System on Chip (SoC) and Internet of Things (IoT) Overview**     * 1. **System on Chip:**   1.1.1 What is System on chip?  1.1.2 Structure of System on Chip.  1.2  **SoC products**:  1.2.1 Field Programmable Gate Array (FPGA)  1.2.2 General Purpose Graphics Processing Units GPU),  1.2.3. Accelerated Processing Unit (APU)  1.2.4. Compute Units.  1.3 **The IoT paradigm giving overview of IoT supported**  **Hardware platforms**  1.3.1. Raspberry pi, SoC on ARM 8 Processors  1.3.2 Arduino and Intel Galileo boards.  1.4 **Network Fundamentals:**  1.4.1. Wired Networking(Router, Switches)  1.4.2. Wireless Networking(Access Points)  **1.5 Introduction to Raspberry Pi:**  1.5.1. Introduction to Raspberry Pi  1.5.2. Raspberry Pi Hardware,  1.5.3. Preparing your raspberry Pi.  **1.6 Raspberry Pi Boot:**  1.6.1. Learn how this small SoC boots without BIOS.  1.6.2.Configuring boot sequences and hardware.  **1.7. Introduction to IoT:**  **1.7.1 IOT concepts**  1.7.1.1 Technologies that led to evolution of IOT  1.7.1.2. IOT and SCADA  1.7.1.3. IOT and M2M  1.7.1.4. IOT and Big Data  1.7.2.Simple IoT LED , Programs.  1.8 **IoT and Protocols**  1.9. **IoT Security:**  1.9.1. HTTP,  1.9.2. UPnp,  1.9.3. CoAP  1.9.4. MQTT  1.9.5. XMPP.  **1.10 IoT Service as a Platform:**  1.10.1. Clayster, Thinger.io,  1.10.2. Sense IoT,  1.10.3. carriots and Node RED.  **1.11 IoT Security and Interoperability:**  1.11.1.Risks,  1.11.2.Modes of Attacks,  1.11.3.Tools for Security and Interoperability. | **20** |
| **2** | **IOT Standards and Components :**  **2.1. Standards:**  2.1.1 Requirement of international standard.  2.2.2 IOT standards in practice.  2.3.3 Operating platforms / Systems.  **2.2. Components of IOT System.**  2.2.1. Design of IOT systems  2.2.2. Development of prototypes.  **2.3. Relevance of IOT for the future.**  2.3.1. IOT in everyday life  2.3.2. Internet of Everything  2.3.3. IOT and Individual Privacy. | **10** |
| **3** | **Advanced IoT Applications and Case Studies**  3.1. **IOT Applications.**  3.1.1. Lighting as a service  3.1.2 Intelligent Traffic systems  3.1.3 Smart Parking  3.1.4 Smart water management  3.1.5 IOT for smart cities  3.2 **IOT in Indian Scenario**  3.2.1 IOT and Aadhaar  3.2.2 IOT for health services.  3.2.3 IOT for financial inclusion.  3.2.4 IOT for rural empowerment.  3.3 **Challenges in IOT implementation**.  3.3.1 Big Data Management.  3.3.2 Connectivity challenges.  3.3.3 Mission critical applications.  3.4 Sensors and sensor Node and interfacing using any  Embedded target boards (Raspberry Pi / Intel Galileo/ARM  Cortex/Arduino) | **18** |
| **Total** | | **48** |

**References:**

a) The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World

b) <http://www.amazon.in/Internet-Things-Smart-citiesChanging/dp/0789754002/ref=sr_1_9?ie=UTF8&qid=1474003280&sr=89&keywords=internet+of+things+book>