GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT (GITAM)
(Deemed to be University, Estd. u/s 3 of UGC Act 1956)
VISAKHAPATNAM ∗HYDERABAD ∗BENGALURU
Accredited by NAAC with 'A^{+}' Grade

REGULATIONS AND SYLLABUS

of

Master of Computer Applications

(w.e.f 2019-20 Admitted batch)

Website: www.gitam.edu

Master of Computer Applications (M.C.A)
REGULATIONS
(W.e.f. 2019-20 admitted batch)

1. ADMISSION

1.1 Admission into M.C.A program of GITAM University is governed by GITAM University admission regulations.

2. ELIGIBILITY CRITERIA

2.1. A pass in BCA or B.Sc. degree with a minimum aggregate of 50% marks / a pass in any degree with minimum aggregate of 50% marks along with Mathematics or Statistics or Computer science as one of the subject.

2.2. Admission into M.C.A (Master Computer Applications) will be based on an All India GITAM Science Admission Test (GSAT) conducted by GITAM University and the rule of reservation, wherever applicable.

3. CHOICE BASED CREDIT SYSTEM

Choice Based Credit System (CBCS) is introduced with effect from the admitted Batch of 2015-16 based on UGC guidelines in order to promote:

- Student Centered Learning
- Cafeteria approach
- Inter-disciplinary learning

Learning goals/ objectives and outcomes are specified leading to what a student should be able to do at the end of the program.

4. STRUCTURE OF THE PROGRAM

4.1 The Program Consists of

i) Foundation Courses (compulsory) which give general exposure to a Student in communication and subject related area.
ii) Core Courses (compulsory).
iii) Discipline centric electives which
   a) are supportive to the discipline
   b) give expanded scope of the subject
   c) give their disciplinary exposure
   d) nurture the student skills
iv) Open electives are of general nature either related or unrelated to the discipline.
v) Practical Proficiency Courses, Laboratory and Project work.

4.2 Each course is assigned a certain number of credits depending upon the number of contact hours (lectures/tutorials/practical) per week.

4.3 In general, credits are assigned to the courses based on the following contact hours per week per semester.

- One credit for each Lecture / Tutorial hour per week.
- One credit for two hours of Practical per week.
- Eight credits for project.

4.4 The curriculum of the Six Semesters M.C.A program is designed to have a total of 128 credits for the award of M.C.A degree.

5. MEDIUM OF INSTRUCTION

The medium of instruction (including examinations and project reports) shall be in English.

6. REGISTRATION

Every student has to register himself / herself for each semester individually at the time specified by the Institute / University.

7. ATTENDANCE REQUIREMENTS

7.1. A student whose attendance is less than 75% in all the courses put together in any semester will not be permitted to attend that end - semester examination and he/she will not be allowed to register for subsequent semester of study. He/she has to repeat the semester along with his / her juniors

7.2. However, the Vice Chancellor on the recommendation of the Principal / Director of the Institute/School may condone the shortage of attendance to the students whose attendance is between 66% and 74% on genuine grounds and on payment of prescribed fee.

8. EVALUATION

8.1. The assessment of the student's performance in a Theory course shall be based on two components: Continuous Evaluation (40 marks) and Semester-end examination (60 marks).

8.2. A student has to secure an aggregate of 40% in the course in continuous and semester end examinations the two components put together to be declared to have passed the course, subject to the condition that the candidate must have secured a minimum of 24 marks (i.e. 40%) in the theory component at the semester-end examination.

8.3. Practical / Viva voce etc. course are completely assessed under Continuous Evaluation for a maximum of 100 marks and a student has to obtain a minimum of 40% to secure Pass Grade. Details of Assessment Procedure are furnished below in Table 1.

Table 1: Assessment Procedure

S. No.	Component of assessment	Marks allotted	Type of Assessment	Scheme of Examination
1	Theory	40	Continuous evaluation	(i) Three mid semester examinations shall be conducted for 15 marks each. The performance in best two shall be taken into consideration. (ii) 5 marks are allocated for quiz. (iii) 5 marks are allocated for assignments.
		60	Semester-end examination	The semester-end examination shall be for a maximum of 60 marks.
	Total	100		
2	Practical	100	Continuous evaluation	60 marks for performance, regularity, record/ and case study. Weightage for each component shall be announced at the beginning of the semester. 40 marks (30 marks for experiment(s) and 10 marks for practical Viva-voce.) for the test conducted at the end of the Semester conducted by the concerned lab Teacher.
	Total	100		
3	Project work	200	Project evaluation	150 marks for evaluation of the project work dissertation submitted by the candidate. 50 marks are allocated for the project Viva-Voce. The project work evaluation and the Viva-Voce shall be conducted by one external examiner outside the University and the internal examiner appointed by the Head of the Department.

9. SUPPLEMENTARY EXAMINATIONS & SPECIAL EXAMINATIONS:

9.1 The odd semester supplementary examinations will be conducted on daily basis after conducting regular even semester examinations in April/May.

9.2 The even semester supplementary examinations will be conducted on daily basis after conducting regular odd semester examinations during November/December

9.3 A student who has completed his/her period of study and still has "F" grade in final semester courses is eligible to appear for Special Examination normally held during summer vacation.

10. PROMOTION TO THE NEXT YEAR OF STUDY

10.1 A student shall be promoted to the next academic year only if he/she completes the academic requirements of 60% of the credits till the previous academic year.

10.2 Whenever there is a change in syllabus or curriculum he/she has to continue the course with new regulations after detention as per the equivalency established by the BoS to continue his/her further studies

11. BETTERMENT OF GRADES

11.1 A student who has secured only a pass or second class and desires to improve his/her class can appear for betterment examinations only in 'n' (where 'n' is no.of semesters of the program) theory courses of any semester of his/her choice, conducted in summer vacation along with the Special Examinations.

11.2 Betterment of Grades is permitted 'only once', immediately after completion of the program of study.

12. REPEAT CONTINUOUS EVALUATION:

12.1 A student who has secured 'F' grade in a theory course shall have to reappear at the subsequent examination held in that course. A student who has secured 'F' grade can improve continuous evaluation marks upto a maximum of 50% by attending special instruction classes held during summer.

12.2 A student who has secured 'F' grade in a practical course shall have to attend Special Instruction classes held during summer.

12.3 A student who has secured 'F' grade in a combined (theory and practical) course shall have to reappear for theory component at the subsequent examination held in that course. A student who has secured 'F' grade can improve continuous evaluation marks upto a maximum of 50% by attending special instruction classes held during summer.

12.4 The RCE will be conducted during summer vacation for both odd and even semester students. Student can register a maximum of 4 courses. Biometric attendance of these RCE classes has to be maintained. The maximum marks in RCE be limited to 50% of Continuous Evaluation marks. The RCE marks are considered for the examination held after RCE except for final semester students.

12.5 RCE for the students who completed course work can be conducted during the academic semester. The student can register a maximum of 4 courses at a time in slot of 4 weeks. Additional 4 courses can be registered in the next slot.

12.6 A student is allowed to Special Instruction Classes (RCE) 'only once' per course.

13. GRADING SYSTEM

13.1 Based on the student performance during a given semester, a final letter grade will be awarded at the end of the semester in each course. The letter grades and the corresponding grade points are as given in Table 2.

Table 2: Grades & Grade Points

Sl.No.	Grade	Grade Points	Absolute Marks
1	O (outstanding)	10	90 and above
2	A+ (Excellent)	9	80 to 89
3	A (Very Good)	8	70 to 79
4	B+ (Good)	7	60 to 69
5	B (Above Average)	6	50 to 59
6	C (Average)	5	45 to 49
7	P (Pass)	4	40 to 44
8	F (Fail)	0	Less than 40
9	Ab. (Absent)	0	-

13.2 A student who earns a minimum of 4 grade points (P grade) in a course is declared to have successfully completed the course, subject to securing an average GPA (average of all GPAs in all the semesters) of 5 at the end of the Program to declare pass in the program.

Candidates who could not secure an average GPA of 5 at the end of the program shall be permitted to reappear for a course(s) of their choice to secure the same.

14. GRADE POINT AVERAGE

14.1 A Grade Point Average (GPA) for the semester will be calculated according to the formula:

$$GPA = \frac{\Sigma [C * G]}{\Sigma C}$$

Where

C = number of credits for the course,
G = grade points obtained by the student in the course.

14.2 To arrive at Cumulative Grade Point Average (CGPA), a similar formula is used considering the student's performance in all the courses taken, in all the semesters up to the particular point of time.

14.3 CGPA required for classification of class after the successful completion of the program is shown in Table 3.

Table 3: CGPA required for award of Class

Class	CGPA Required
First Class with Distinction	≥ 8.0*
First Class	≥ 6.5
Second Class	≥ 5.5
Pass Class	≥ 5.0

* In addition to the required CGPA of 8.0 or more the student must have necessarily passed all the courses of every semester in first attempt.

15. ELIGIBILITY FOR AWARD OF THE M.C.A DEGREE

15.1 Duration of the program: A student is ordinarily expected to complete M.C.A program in six semesters of three years. However, a student may complete the program in not more than Five years including study period.

15.2 However the above regulation may be relaxed by the Vice Chancellor in individual cases for cogent and sufficient reasons.

15.3 A student shall be eligible for award of the M.C.A Degree if he / she fulfills all the following conditions.

a) Registered and successfully completed all the courses and projects.
b) Successfully acquired the minimum required credits as specified in the curriculum corresponding to the branch of his/her study within the stipulated time.
c) Has no dues to the Institute, hostels, Libraries, NCC / NSS etc, and
d) No disciplinary action is pending against him / her.

15.4 The degree shall be awarded after approval by the Academic Council.

16. DISCRETIONARY POWER:

Not with standing anything contained in the above sections, the Vice Chancellor may review all exceptional cases, and give his decision, which will be final and binding.

Master of Computer Applications (M.C.A.)
Scheme of Instruction
I SEMESTER

Sl. No.	Course Code	Name of the Course	Credits	Scheme of Instruction		Total	Scheme of Examination		
				Hours per Week			Duration in Hrs.	Maximum Marks	
				L/T	P			Sem. End Exam	Con. Eval
1	SCA 601	Object oriented programming with C++	4	4	0	4	3	60	40
2	SCA 603	Computer Organization	4	4	0	4	3	60	40
3	SCA 605	Web Programming	4	4	0	4	3	60	40
4	SCA 607	Mathematics for Computer Science	4	4	0	4	3	60	40
PRACTICALS :									
1	SCA 621	Programming with C++ Lab	2	0	4	4	3	--	100
2	SCA 623	Web Programming Lab	2	0	4	4	3	--	100
3	SSE 601	Data Visualization Techniques	2	3	0	3	--	--	100
		Total	22	19	8	27		240	460

M.C.A. – II SEMESTER

Sl. No.	Course Code	Name of the Course	Credits	Scheme of Instruction		Total	Scheme of Examination		
				Hours per Week			Duration in Hrs.	Maximum Marks	
				L/T	P			Sem. End Exam	Con. Eval
1	SCA 602	Database Management System	4	4	0	4	3	60	40
2	SCA 604	Data Structures using C++	4	4	0	4	3	60	40
3	SCA 606	Probability and Statistics	4	4	0	4	3	60	40
4	SCA 608	Operating Systems	4	4	0	4	3	60	40
5	SAE 604	Technical Communication Skills	3	4	0	3	3	60	40
PRACTICALS :									
1	SCA 622	Database Management System Lab	2	0	4	4	3	--	100
2	SCA 624	Data Structures using C++ Lab	2	0	4	4	3	--	100
3	SCA 626	Programming with UNIX Lab	2	0	4	4	3	--	100
		Total	25	20	12	32	--	300	500

M.C.A. – III SEMESTER

Sl. No	Course Code	Name of the Course	Credits	Scheme of Instruction		Total	Scheme of Examination		
				Hours per Week			Duration in Hrs.	Maximum Marks	
				L/T	P			Sem. End Exam	Con. Eval
1	SCA 701	Programming with JAVA	4	4	0	4	3	60	40
2	SCA 703	Design and Analysis of Algorithms	4	4	0	4	3	60	40
3	SCA 705	Optimization Techniques	4	4	0	4	3	60	40
4	SCA 707	Software Engineering	4	4	0	4	3	60	40
5	SOE XXX	Open Elective	3	3	0	3	3	60	40
PRACTICALS :									
	SCA721	Programming with JAVA Lab	2	0	4	4	3	--	100
	SCA 723	MEAN Stack Lab	2	0	4	4	3	--	100
	SCA 791	Minor Project	2	0	--	--	--	--	100
		Total	25	19	8	25		300	500

M.C.A. – IV SEMESTER

Sl. No.	Course Code	Name of the Course	Credits	Scheme of Instruction		Total	Scheme of Examination		
				Hours per Week			Duration in Hrs.	Maximum Marks	
				L/T	P			Sem. End Exam	Con. Eval
1	SCA702	Computer Networks	4	4	0	4	3	60	40
2	SCA704	Data Mining	4	4	0	4	3	60	40
3	SCA706	Artificial Intelligence	4	4	0	4	3	60	40
4	SCA708	Python Programming	4	4	0	4	3	60	40
5	SCA 742 SCA 744 SCA 746	**Generic Elective – I** Software Testing Internet of Things Enterprise Resource Planning	4	4	0	4	3	60	40
PRACTICALS :									
	SAE 722	Technical Communication Skills Lab	2	0	4	4	3	--	100
	SCA 722	Data Mining Lab using R	2	0	4	4	3	--	100
	SCA 724	Python Programming Lab	2	0	4	4	3	--	100
		Total	26	20	12	32		300	500

M.C.A. – V SEMESTER

Sl. No.	Course Code	Name of the Course	Credits	Scheme of Instruction		Total	Scheme of Examination		
				Hours per Week			Duration in Hrs.	Maximum Marks	
				L/T	P			Sem. End Exam	Con. Eval
1	SCA801	Cloud Computing	4	4	0	4	3	60	40
2	SCA803	Cyber Security	4	4	0	4	3	60	40
3	SCA805	Machine Learning	4	4	0	4	3	60	40
4	SCA841 SCA843 SCA845	**Generic Elective – II** DevOps Big Data Analytics J2EE Technologies	4	4	0	4	3	60	40
PRACTICALS :									
	SCA881 SCA883 SCA885	**Generic Elective – II Lab** DevOps Lab Big Data Lab J2EE Technologies Lab	2	--	4	4	3	--	100
	SCA 887	Microsoft Power BI Lab	2	--	4	4	3	--	100
	SCA891	Summer Internship	2	--	--	--	--	--	100
		Total	22	16	8	24		240	460

M.C.A. – VI SEMESTER

Sl. No.	Course Code	Name of the Course	Credits	Scheme of Instruction		Total	Scheme of Examination		
				Hours per Week			Duration in Hrs.	Maximum Marks	
				L/ T	P			Sem. End Exam	Con. Eval
1	SCA 892	Project Work	8	--	--	--	--	50	150
		Total	8	--	--	--	--	50	150

Total Credits: 22+25+25+26+22+8 = 128

M.C.A.- I SEMESTER

SCA 601 OBJECT ORIENTED PROGRAMMING WITH C++

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Object Oriented Programming (OOP) is a programming model where programs are organized around objects and data rather than action and logic. OOP allows decomposition of a problem into a number of entities called objects and then builds data and functions around these objects. The major purpose of C++ programming is to introduce the concept of object orientation to the C programming language. C++ is a statically-typed, free-form, (usually) compiled, multi-paradigm, intermediate-level general-purpose middle-level programming language.

Course Objectives:

- To enable the student to understand the importance of Principles of Object Oriented Programming
- To learn the basic concepts and benefits of OOP, Application of OOP, OOP languages
- To understand the structure and basic concepts of C++ Programming language
- To practice functions, classes and objects in C++ Programming language
- To demonstrate constructor, destructors, operator overloading and type conversions
- To make the student learn inheritance, extending classes, pointers, virtual functions and polymorphism and templates.
- To explain managing console, I/O operations, working with files and exception handling

UNIT – I

Principles of Object Oriented Programming: Software Crisis, Software Evolution, A look at Procedure Oriented Programming, Object Oriented Programming Paradigms, Basic concepts and benefits of OOP, Application of OOP, OOP languages. Structure of C++ Program, Applications of C++, Tokens, Keywords, Identifiers and Constants, Basic data types, User Defined data types, Storage Classes, Derived Data Types, Symbolic constants, Type Compatibility, Declaration of Variables, Dynamic initialization, Reference variables, Operators, Scope resolution, Member Dereferencing Operators, Memory Management Operators, Manipulators, Type Casting, Expressions and their types, Special Assignment Expressions, Implicit Conversions, Operator Overloading, Operator Precedence, Control Structures. (12)

Learning Outcomes:

By the end of the unit the student will be able to

- Learn the principles of Object Oriented Programming.
- Understand the basic concepts and benefits of OOP, Application of OOP and OOP languages.
- Practice the structure of C++ Program and applications of C++.
- Know the data types, variables, operators, and expressions.
- Demonstrate the Operator Overloading, Operator Precedence, and Control Structures.

UNIT – II

Functions: Main Function, Function Prototyping, Call by Reference, Return By Reference, Inline functions, Default arguments, Const Arguments, Recursion, Function overloading, Friend and virtual functions, Math Library Functions**.**

Classes and Objects: Class specification, Member function definition, Making outside function inline, Nesting Of Member Functions, Private Member Function, Arrays Within a Class, Memory Allocation For Objects, Static Data Members And Member Functions, Array of Objects, Objects As Arguments, Friend Functions, Returning Objects, const Member Function, Pointer To Members, Local Classes.

Learning Outcomes:

By the end of the unit the student will be able to

- Understand the functions.
- Differentiate the Call by Reference, and Return By Reference.
- Learn the Inline functions, Default arguments, and const Arguments.
- Practice Recursion, Function overloading, Friend and virtual functions, and Math Library Functions.
- Demonstrate the Classes and Objects.
- Understand the Member Function, Pointer To Members, and Local Classes.

UNIT – III

Constructors and Destructors: Constructors, Parameterized Constructors, Multiple Constructors in a Class, Constructors with Default Arguments, Dynamic initialization of Objects, Copy constructors, Dynamic Constructors, Constructing Two Dimensional Arrays, const Objects, Destructors

Operator Overloading and Type Conversions: Defining Operator Overloading, Overloading Unary And Binary Operator, Overloading Binary Operators Using Friends, Manipulation Of String Using Operators, Rules For Overloading Operators, Type Conversion (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Learn the Constructors and Destructors.
- Understand the Construction of Two Dimensional Arrays.
- Know the Operator Overloading and Type Conversions.
- Practice the Manipulation of Strings.

UNIT – IV

Inheritance Extending Classes: Introduction, Defining Derived Classes, Single Inheritance, making a Private Member Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance, Virtual Base Classes, Abstract Classes, Constructors In Derived Classes, Member Classes, Nesting of Classes.

Pointers, Virtual Functions and Polymorphism: Pointers, Pointer to objects, this pointer, Pointer to Derived Classes, Virtual Functions, Pure Virtual Functions, Virtual Constructors and Destructors.

Templates: Introduction, Class Template, Class Template With Multiple Parameters, Function**,** Templates, Function Templates With Multiple Parameters, Overloading of Template Functions, Member Function Templates, Non type Template Arguments (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Understand the Inheritance.
- Differentiate the Virtual Base Classes and Abstract Classes.
- Practice Pointers, Virtual Functions and Polymorphism.
- Know Virtual Constructors and Destructors.
- Learn Templates.

UNIT – V

Managing Console, I/O Operations: Streams in C++, Stream Classes, Unformatted I/O Operations, Formatted Console I/O Operations, Managing Output With Manipulators.

Working with Files: Introduction, Classes For File Stream Operations, Opening And Closing A File, Detecting End-Of-File, File Modes, File Pointers And Their Manipulations, Sequential Input and Output Operations, Updating a File: Random Access, Error Handling During File Operations, Command Line Arguments.

Exception Handling: Introduction, Basics Of Exception Handling, Exception Handling Mechanism, Throwing And Catching Mechanism, Rethrowing an Exception, Specifying Exceptions, Exceptions In Constructors And Destructors, Exception In Operator Overloaded functions. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Understand the Managing Console I/O Operations
- Practice working with Files
- Develop Exception Handling
- Demonstrate throwing, catching and rethrowing an exceptions

Text Books:

1. Object Oriented Programming with C++ by E.Balagurusamy, Tata MC GrawHill, 6th edition, 2013.

Reference Books:

1. Mastering C++ by Venugopal K R, Rajkumar Buyya , Tata Mc Graw Hill, 2nd edition, 2013.

2. C++ Programming Language by Bjarne Stroustrup, Addison-Wesley Professional, 4th edition, 2013.

3. C++ Primer by Barbara E Moo, Stanley B. Lippman , Josee Lajoie, Pearson Education, 4th edition, 2007.

Course Outcomes:

- Able to understand the basic concepts and benefits of OOP
- Able to know the Principles of Object Oriented Programming
- Able to demonstrate on class, object, functions, constructors and pointers
- Able to practice the File concepts
- Able to learn template concepts
- Understand the Console I/D operations

Prepared by: Mr. V.S.Prasad Babu Verified by : Prof. V. NagaLakshmi

M.C.A.-I SEMESTER
SCA 603 COMPUTER ORGANIZATION

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Computer organization and architecture mainly focuses on various parts of the computer in order to reduce the execution time of the program, improve the performance of each part. Computer Organization is study of the system from software point of view and gives overall description of the system and working principles without going into much detail. This course is a study of the system from hardware point of view and emphasis on how the system is implemented.

Course Objectives:

- To enable the student on basic knowledge of a Digital Computer, its architecture, components and their organization.
- To understand Digital Logic Circuits and Digital Components.
- To learn Register Transfer and Micro operations.
- To understand Basic Computer Organization and Design.
- To learn about Central Processing Unit and Input-output organization.
- To know Memory Organization.

UNIT – I

Digital Logic Circuits: Digital Computers, Logic gates, Boolean algebra, Map simplification, Combinational circuits, Flip-flops, Sequential circuits.

Digital Components: Integrated circuits, Decoders, Multiplexers, Registers, Shift registers, Binary counters, Memory unit.

Data Representation: Data types, Compliments, Fixed-point representation, Floating point representation. (12)

Learning Outcomes:

By the end of the unit the student will be able to

- Understand Digital Computers, Logic gates, Boolean algebra and Map simplification.
- Follow Combinational circuits, Flip-flops and Sequential circuits.
- Practice Integrated circuits, Decoders, Multiplexers, Registers, Shift registers, Binary counters and Memory unit.
- Know Data types and Compliments, Fixed-point representation and Floating point representation.

UNIT – II

Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus and memory transfers, Arithmetic operations, Logic micro operations, Shift micro operations, Arithmetic Logic and Shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers, Computer instructions, Timing and control, Instruction cycle, Memory-Reference instructions, Input-output and interrupt, Design of basic computer, Design of accumulator logic. (12)

Learning Outcomes:

By the end of the unit the student will be able to

- Compare Register Transfer Language, Register Transfer, Bus and memory transfers and Arithmetic operations.
- Understand Logic micro operations ad Shift micro operations, Arithmetic Logic and Shift unit.
- Learn Instruction codes, Computer Registers, Computer instructions, Timing and control, Instruction cycle, Memory-Reference instructions.
- Know Input-output and interrupt, Design of basic computer and Design of accumulator logic.

UNIT – III

Central Processing Unit: General Register Organization, Stack organization, Instruction formats, Addressing Modes, Data Transfer and Manipulation, Program control. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Learn General Register Organization, and Stack organization
- Understand Instruction formats, Addressing Modes
- Explain Data Transfer and Manipulation
- Identify Program control

UNIT – IV

Input-output organization: Peripheral devices, Input-output interface, Asynchronous data transfer, Modes of transfer, Priority interrupt, DMA. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Classify Peripheral devices, and Input-output interface.
- Find Asynchronous data transfer.
- Illustrate Modes of transfer.
- Understand Priority interrupt and DMA.

UNIT – V

Memory Organization: Memory hierarchy, Main memory, Auxiliary memory, Associative memory, Cache memory, Virtual memory, Memory management hardware. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Understand Memory hierarchy.
- Differentiate Main memory, and Auxiliary memory.
- Compare Associative memory and Cache memory.
- Learn Virtual memory and Memory management hardware.

Text books:

1. Computer System Architecture by M. Morris Mano, 3rd edition, Pearson Prentice hall, tenth impression 2013.

Reference books:

1. Computer System Architecture and Organization by M.Usha and T.S.Srikanth, Wiley India Private Limited, 2012.
2. The Basics of Computer System Architecture by Gursharan Singh and Maninder kaur, Modern publishers.

Course Outcomes:

- Learn the basics of a Digital Computer, its architecture, components and their organization.
- Understand the hardware and upcoming processor architecture and its evolution with change in working style.
- Find the Digital Logic Circuits and Digital Components.
- Know the Basic Computer Organization and Design.
- Summarize about Central Processing Unit.
- Understand the functionality of Memory Organization.

Prepared by: Mr. V.S.Prasad Babu Verified by : Prof. V. NagaLakshmi

M.C.A.- I SEMESTER
SCA 605 WEB PROGRAMMING

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Web Programming course provides a deep dive into the World Wide Web, and the underlying structures that needs to be understood to build robust applications and websites that are efficient and easy to maintain. Help the students to learn how HTML and DHTML documents provide the foundation for CSS and JavaScript, and see how to layer presentation and interactions on top.

Course Objectives:

- To enable the student to understand the importance of Web Programming.
- To learn the features and usage of HTML tags in Web Programming.
- To learn the DHTML advantage in writing Web pages.
- To learn the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts.
- To write programmatic access to objects within both the client application and other applications using Java Script.
- To learn how web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes using AJAX.

UNIT -I

Inserting Images, Creating Image Links; Horizontal Rules: Changing the Height of a Horizontal Rule, Changing between Shaded and Un-shaded Horizontal Rule, Changing the Width of a Horizontal Rule, Setting the Alignment of a Horizontal Rule; Address Tag; Working with Text: Text Alignment, Changing Font Sizes and Colors: Setting Font Sizes, Setting the Base Font, Using the Small and Big tags, Changing the Font Color; Using a Background Image; Marquee TagTables Images, Frames, Forms, Type Attribute: Check box, Hidden, Image, Radio, Reset, Submit, Text; Other <INPUT> attributes: Value, SRC, Checked, Size, Max length, Align, Select tag, Text Area, CGI, Get, Post

Learning Outcomes:

By the end of the unit the student will be able to

- Insert Images in a web page.
- Create Image Links; Horizontal Rules: Change the Height of a Horizontal Rule, Change between Shaded and Un-shaded Horizontal Rule, Change the Width of a Horizontal Rule.
- Set the Alignment of a Horizontal Rule; Address Tag; Work with Text: Text Alignment, Changing Font Sizes and Colors: Set Font Sizes, Set the Base Font, Use the Small and Big tags, Change the Font Color.
- Use a Background Image; Marquee TagTables Images, Frames, Forms, Type Attribute: Check box, Hidden, Image, Radio, Reset, Submit, Text; Other <INPUT> attributes: Value, SRC, Checked, Size, Max length, Align, Select tag, Text Area, CGI, Get, Post.

UNIT –II

Cascading Style Sheets (CSS): Properties Table: Using the style Attribute, Creating Classes and IDs, Generating External Style Sheets, Typography, Consistency, Types of styles, Specifying class within HTML document, Style placement: Inline style, Span & div tags, header styles, Text and font attributes: Font Vs CSS, changing fonts, text attributes, Advance CSS properties: Backgrounds, Box properties and Positioning

Learning Outcomes:

By the end of the unit the student will be able to

- Use Properties Table Using the style Attribute, Creating Classes and IDs.
- Generate External Style Sheets, Typography, Consistency, Types of styles, and Specifying class within HTML document.
- Use of Style placement, Inline style, Span & div tags, header styles, Text and font attributes.
- Understand Font Vs CSS, changing fonts, text attributes, and Advance CSS properties: Backgrounds, Box properties and Positioning.

UNIT –III

DHTML: DHTML Overview & Definitions, Dynamic Images, Image Rollovers, Slide Shows, Dueling DOMs, The Document Object Model, The Navigator 4.x DOM, The Internet Explorer DOM, Dealing with DOM Differences, Creating the Core DHTML Library, The Custom Object Technique, Adding Methods to a Custom Object, Adding Secondary Methods and Properties & Active Element Object, Moving Elements on the Page, Moving in Geometric Shapes

Learning Outcomes:

By the end of the unit the student will be able to

- Understand the Dynamic Images, Image Rollovers, Slide Shows, Dueling DOMs, The Document Object Model, The Navigator 4.x DOM, The Internet Explorer DOM, Dealing with DOM Differences.
- Create the Core DHTML Library, The Custom Object Technique, and adding Methods to a Custom Object.
- Add Secondary Methods and Properties & Active Element Object, Moving Elements on the Page, and moving in Geometric Shapes.

UNIT –IV

JavaScript Programming: Introduction to JavaScript: Utility of JavaScript, Evolution of the JavaScript Language, JavaScript Versions and Browser Support, Differences Between Client-Side vs. Server-Side JavaScript, Statements and Operators, Variable Declarations, Assignment Operators and Statements, Arithmetic Operators, Logical Operators, Comparison Operators, String Operators, Conditional Operators, Operator Precedence; Implementing Control Constructs: Conditional and Looping Constructs, if else Statements, do while Statements, for in Statement, switch Statement; Implementing Functions: Defining Functions, Calling Functions, Passing Arguments, Local vs. Global Variables, Using the Return Statement, Nested Functions; JavaScript Objects: The JavaScript Object Model and Hierarchy, JavaScript Object Properties, Object Methods, New Keyword, This Keyword, Creating New Object Instances Using Constructor Functions, String, Date and Array Objects, Construction of Custom Objects with Individual Properties and Methods

Learning Outcomes:

By the end of the unit the student will be able to

- Differentiate between Client-Side vs. Server-Side JavaScript.
- Learn the Java Script language.

- Practice the JavaScript Object Model and Hierarchy, JavaScript Object Properties.
- Understand the Object Methods, New Keyword, This Keyword, Creating New Object Instances Using Constructor Functions, String, Date and Array Objects, Construction of Custom Objects with Individual Properties and Methods.

UNIT –V

Introduction to AJAX : The Purpose of Ajax, Traditional Web Application, An Ajax Web Application, The XMLHttpRequest, Object Creating an XMLHttpRequest, Object Using an XMLHttpRequest ,Object Handling the Response, AJAX Advantages and Disadvantages, AJAX components, Key Elements of AJAX, The Purpose of Frameworks, Choosing a Framework, Dojo , Downloading Dojo, Using Dojo for Ajax, Prototype, Downloading Prototype, Using Prototype for Ajax

Learning Outcomes:

By the end of the unit the student will be able to

- Understand the purpose of AJAX.
- Find the Ajax Web Application, The XMLHttpRequest, Object Creating an XMLHttpRequest, Object Using an XMLHttpRequest, and Object Handling the Response.
- Compare the AJAX Advantages and Disadvantages.
- Identify the Purpose of Frameworks, choosing a Framework, Dojo, Downloading Dojo, Using Dojo for Ajax, Prototype, Downloading Prototype, and using Prototype for Ajax.

Text Book:

1. Web Enable Commercial Application Development Using … HTML, Javascript, DHTML and PHP by Ivan Bayross, BPB Publications, 4th revised edition, 2010 .
2. Ajax: The Complete Reference Thomas A. PowellMcGraw Hill Education (20 March 2008)

Reference Books:

1. Complete Reference HTML, T. A. Powell, 3rd edition, TMH, 2003.
2. The Complete Reference - PHP by Steven Holzner, Tata McGraw Hill, 2008.
3. Web Technology and Design, Xavier, C, New Age International, 2013.

Course Outcomes:

- Able to understand the Web Programming development using HTML.
- Learn the HTML tags and functionality and it role in web development.
- Identify the CSS role in Web Development and writing CSS.
- Illustrate DHTML usage in web development and DHTML tags and its functionality.
- Practice the Java Script usage in web development and writing different validation scripts using Java Script.
- Know the purpose of AJAX in Web development and key elements of AJAX and its functionality.

Prepared by: Mr. V.S.Prasad Babu Verified by : Prof. V. NagaLakshmi

M.C.A. - I SEMESTER
SCA 607 MATHEMATICS FOR COMPUTER SCIENCE

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:
Mathematics for computer Science is introduced to impart knowledge of basic mathematical concepts and numerical methods. This course helps to understand the concepts and results in Mathematical Logic, Set Theory, Lattices and Boolean Algebra, Graph Theory and Numerical methods.

Course Objectives:

- To understand the difference between primitive statement and compound statement
- To learn the basic concept and applications of theory of inference for the statement calculus and predicate calculus
- Ability to define a set , relation and a function with their properties
- Ability to implement features of lattices and Boolean algebra.
- To understand the concept of graphs , directed graphs, and trees
- To solve numerically algebraic and transcendental Equations and numerical integration problems.

UNIT - I

Mathematical Logic: Statements and Notation, Connectives, Normal Forms, The Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus. (10)

Learning Outcomes:
By the end of this Unit, the student will be able to

- Understand the connectives used for statements
- Explain disjunctive normal form and conjunctive normal form
- Explain direct method to solve inference theory statements
- Symbolize the predicate statements
- Extend the inference theory of statement calculus to theory of the predicate calculus

UNIT - II

Set Theory: Basic Concepts of Set Theory, Relations and Ordering.
Functions: Definitions and Introduction, Composite of Functions, Inverse Functions, Binary and n-ary Operations, Characteristic Function of a Set. (10)

Learning Outcomes
By the end of this Unit, the student will be able to

- Define various sets, various relations, and various orderings
- Compare different types of functions and their properties
- Practice functions
- Analyze characteristics function of a given set

UNIT - III

Lattices and Boolean Algebra: Lattices: Definition and Examples, Properties of Lattices, Sub lattices, Direct Product and Homomorphism, Some Special Lattices.
Boolean Algebra: Definition and Examples, Sub algebra, Direct Product and Homomorphism, Boolean Functions. (10)

Learning Outcomes
By the end of this Unit, the student will be able to

- Define the lattice and its properties

- Explain the Hasse diagram
- Extend the concept of lattice to Boolean algebra
- Outline the different types of lattices
- Define Boolean algebra and Boolean functions

UNIT - IV

Graph Theory: Graphs, Multi graphs, Directed Graphs, Complete, Regular and Bipartite Graphs, Planar Graphs, Tree Graphs, Labeled and Weighted Graphs, Basic Definitions, Sub-graphs, Isomorphic Graphs, Paths, Connectivity, The Bridges of Konigsberg, Traversable Multi-graphs. Rooted Trees, Sequential Representation of Directed Graphs, Warshall's Algorithm and Shortest Path (Minima) Algorithm only. (10

Learning Outcomes

By the end of this Unit, the student will be able to

- Explain the concept of graph, directed graph, and trees
- Summarize different types of graphs
- Define various types of traversable graphs
- Explain the concept of rooted trees
- Explain the Warshall's algorithm to find path matrix with an example
- Explain Shortest path algorithm with an example

UNIT - V

Numerical Methods: Solution of Algebraic and Transcendental Equations: The Bisection Method, The Methods of False Position, The Iteration Methods, Newton-Raphson Method.

Numerical Integration: Trapezoidal Rule, Simpson's 1/3 Rule, Simpson's $3/8^{th}$ Rule. (10)

Learning Outcomes

By the end of this Unit, the student will be able to

- Compare various methods to solve algebraic and transcendental equations
- Evaluate numerically solution of an equation using bisection method, iteration methods
- Compare methods using methods of false position and Newton-Raphson
- Apply the concept of trapezoidal rule, simpson's rule in numerical integration

Text Books:

1. Discrete Mathematical Structures with applications to computer science by J.P. Tremblay and R. Monohar, Tata McGraw – Hill.
2. Discrete Mathematics by Seymour Lipschutz and Marc Lipson, Schaum's outlines, Tata McGraw Hill.
3. Introductory Methods of Numerical Analysis by S.S Sastry, Prentice – Hall India.

Reference Books:

1. Discrete Mathematics and its Applications by Kenneth H.Rosen, Tata McGraw – Hill.
2. Numerical Methods for Engineers by Steven C. Chopra and Raymond P.Canale, Mc Graw Hill

Course Outcomes:

- Able to understand the difference between primitive statement and compound statement
- Learn the basic concepts and applications of theory of inference for the statement calculus and predicate calculus
- Define the lattice and Extend the concept of lattice to Boolean algebra
- Able to understand the concept of graphs , directed graphs, and trees
- Able to solve numerically algebraic and transcendental Equations and numerical integration problems.

Prepared by: Prof. N. Ravi Shankar Verified by : Prof. V. NagaLakshmi

M.C.A.- I SEMESTER
SCA 621 PROGRAMMING WITH C++ Lab

Hours per week: 3 Examination: 100 Marks

Credits: 2

1. Program to perform the input and output statements.
2. C++ program to demonstrate Functions.
3. C++ program to implement Class concepts.
4. C++ program to implement Member functions.
5. C++ program to demonstrate Array of objects.
6. C++ program to demonstrate Constructors, parameterized Constructors.
7. C++ program to demonstrate copy constructors.
8. C++ program to demonstrate Friend function.
9. C++ program to implement Inline function.
10. C++ program to demonstrate Function Overloading.
11. C++ program to implement Operator Overloading (using unary and binary operators).
12. C++ program to demonstrate Operator Overloading using Friend Concept.
13. C++ program to implement Inheritance.
14. C++ program to implement Function overriding.
15. C++ program to demonstrate constructors in derived and base class.
16. C++ program to implement Virtual base class and abstract base class.
17. C++ program to demonstrate File handling.
18. C++ program to demonstrate streams and manipulators.
19. C++ program to demonstrate Exception Handling mechanism.
20. C++ program to demonstrate Class Templates and Function Templates.

Reference Book:

1. Object Oriented Programming with C++ by E.Balagurusamy, Tata McGraw Hill Education, 4th edition, 2008.

Course Outcomes:

- Understand the input and output statements and Functions.
- Implement Class concepts and Member functions.
- Demonstrate Array of objects, Constructors and parameterized Constructors.
- Demonstrate copy constructors and Friend function, Inline function and Function Overloading.
- Implement Operator Overloading using unary and binary operators and Operator Overloading using Friend Concept.
- Implement Inheritance and Function overriding, constructors in derived and base class, Virtual base class and abstract base class.

Prepared by: Mr. V.S.Prasad Babu Verified by : Prof. V. NagaLakshmi

M.C.A.- I SEMESTER
SCA 623 WEB PROGRAMMING LAB

Hours per week: 3 Examination: 100 Marks

Credits: 2

1. Write a html program for Creation of web site with forms, frames, links, tables etc
2. Design a web site using HTML and DHTML. Use Basic text Formatting, Images,
3. Create a script that asks the user for a name, and then greets the user with "Hello" and the user name on the page.
4. Creating a Web Page using Image Map
5. Write a webpage that displays college information using various style sheet Style Sheets
6. Write a DHTML code for creating the web page for validating the web form
7. Create a script that collects numbers from a page and then adds them up and prints them to a blank field on the page.
8. Create a script that prompts the user for a number and then counts from 1 to that number displaying only the odd numbers.
9. Create a script that will check the field in Assignment 1 for data and alert the user if it is blank. This script should run from a button.
10. Using CSS for creating web sites
11. Develop and demonstrate a DHTML file that includes JavaScript script for the following problems: a) Input: A number n obtained using prompt Output: The first n Fibonacci numbers b) Input: A number n obtained using prompt Output: A table of numbers from 1 to n and their squares using alert
12. a) Develop and demonstrate, using JavaScript, a DHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected. b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8)
13. a) Develop and demonstrate, using JavaScript script, a DHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible. b) Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom.
14. Explore the following application in AJAX: Searching in real time with live Searches, Getting the answer with auto complete, Chatting with friends ,Dragging and dropping with Ajax, Getting instant login feedback, Ajax-enabled popup menus, Modifying Web pages on the fly.
15. Write a program to change content of web page using AJAX.
16. Write a program to change content of web page using AJAX.

Reference books:

1. The Animate CC, The official training workbook from Adobe. Russel Chun 2017.
2. Photoshop CC: The missing manual, Lesa Snider, O'Reilly Media.
3. Flash CS6/CS5/CS4 in Simple Steps, Kogent Learning Solution Inc. Wiley.
4. Macromedia Flash MX: A Beginner's Guide, Brian Underdahl, MGH.

Course Outcomes:

- Design a website using HTML and DHTML.
- Able to write a DHTML code for creating the web page for validating the web form.
- Able to practice a script that prompts the user for a number and then counts from 1 to that number displaying only the odd numbers.
- Demonstrate CSS for creating web sites.
- Practice a DHTML file that includes Javascript.
- Develop and demonstrate, using Javascript script, a DHTML document that the USN and contains three short paragraphs of text.

Prepared by: Mr. V.S.Prasad Babu Verified by : Prof. V. NagaLakshmi

M.C.A. – I SEMESTER

SSE 601 DATA VISUALIZATION TECHNIQUES

Hours per week: 3 Examination: 100Marks

Credits: 2

1. Program to Demonstrate the Data Sources, Custom Data View, Extracting Data
2. Program to Demonstrate the Fields Operations, Editing Metadata
3. Program to Demonstrate the Data Joining, Data Blending
4. Program to Demonstrate the Worksheets
5. Program to Demonstrate the Add Worksheets, Rename Worksheet, Save & Delete Worksheet, Reorder Worksheet, Paged Workbook
6. Program to Demonstrate the Calculations
7. Program to Demonstrate the Operators
8. Program to Demonstrate the Functions
9. Program to Demonstrate the Numeric Calculations, String Calculations
10. Program to Demonstrate the Date Calculations, Table Calculations, LOD Expressions
11. Program to Demonstrate the Sort & Filters, Basic Sorting, Basic Filters
12. Program to Demonstrate the Quick Filters, Context Filters, Condition Filters, Top Filters
13. Program to Demonstrate the Charts ,Bar Chart, Line Chart, Pie Chart, Crosstab, Scatter Plot
14. Program to Demonstrate the Bubble Chart, Bullet Graph, Box Plot, Tree Map, Bump Char, Gantt Chart
15. Program to Demonstrate the Histogram, Motion Charts, Waterfall Charts

Text Book:

1. Murray, Daniel G. (2014), Tableau your data: Fast and Easy Visual Analysis with Tableau Software, New Delhi: Wiley India.

Reference Books:

1. Milligan, N.J. (2015), Learning Tableau, Mumbai: PACKT / Shroff Publishers.
2. Ferrari, Alberto., Russo, Marco. (2016), Introducing Microsoft Power BI, Microsoft Press (Free ebook available: https:// blogs.msdn.microsoft.com /microsoft_press/2016/06/16/free-ebook-introducing-microsoft-power-bi/)
3. Jones, B. (2014), Communicating Data with Tableau, Mumbai: PACKT / Shroff Publishers
3. Collie, Rob., Singh, Avichal.(2016), Power Pivot and Power BI, Holy Macro Books.

Course Outcomes:

- Able to learn how to customize the data
- Able to understand how to extract the data
- Able to perform field operations and editing meta data
- Able to practice on worksheets
- Demonstrate different functions and calculations
- Practices on different charts and histograms

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A.-II SEMESTER
SCA602 DATABASE MANAGEMENT SYSTEM

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Database Management System course is intended to deliver students the elementary concepts of a database management system and make them to design E-R and implement a database application built over those concepts. It also introduces advanced level areas like transaction processing, concurrency control and recovery management

Course Objectives:

- To learn introductory concepts of databases
- To understand the database system concepts and architecture
- To develop conceptual modelling
- To design relational data model
- To demonstrate the creation, altering and modification of database with SQL
- To Analyze the concepts of transaction processing, concurrency control and database recovery

UNIT - I

Introduction and Conceptual Modeling, Databases and Database Users: Introduction, Characteristics of Database Approach, Actors on the Scene, Workers behind the Scene, Advantages of using DBMS Approach.

Database System, Concepts and Architecture: Data Models, Schemas and Instances, Three Schema Architecture and Data Independence, Database Language and Interfaces, The Database System Environment, Centralized and Client/Server Architecture of Database Management Systems, Classification of Database Management Systems. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn the introductory concepts of databases
- Analyze the characteristics of database approach
- Understand the database system environment
- Know the three schema architecture
- Find the different database architectures

UNIT - II

Data Modeling Using The ER Model: High Level Conceptual Data Models for Database Design, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraint, Weak Entity Types, ER Diagrams, Naming Conventions and Design Issues. The Enhanced Entity Relationship modeL, UML Class Diagrams, Relationship Types of Degree Higher Than Two.

The Relational Data Model and Relational Database Constraints: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions and Dealing with Constraints Violations. (12)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn introduction to Data Models and conceptual modeling
- Understand the introductory concepts: Entity, Attributes and Relationship
- Draw an Entity Relationship diagram
- Know the UML class diagrams
- Summarize relational model concepts, constraints and database schema

UNIT – III

The Relational Algebra: Unary Relational Operations - SELECT and PROJECT, Relational Algebra Operations from Set Theory, Binary Relational Operations - JOIN and DIVISION; Additional Relational Operations, Examples of Queries in Relational Algebra.

Relational Database Design By ER And EER To Relational Mapping: Relational Database Design using ER to Relational Mapping, Mapping EER Model Constructs to Relations. (8)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn relational algebra operations
- Understands Unary and Binary operations
- Develop relational algebra expressions
- Know the concepts of JOIN and DIVISION
- Transform ER to Relational Model

UNIT – IV

SQL-Schema Definition, Constraints, Queries and Views: SQL Data Definition and Data types, Specifying Constraints in SQL, Schema Change Statements in SQL, Basic Queries in SQL, More Complex SQL Queries, INSERT, DELETE, UPDATE Statements in SQL, Additional Features of SQL, Specifying Constraints As Assertions and triggers, Views, Additional features of SQL.

Functional Dependencies and Normalization for Relational Databases: Informal Design Guidelines for Relational Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of 2nd and 3rd Normal Forms, Boyce-Codd Normal Form

Relational Database Design Algorithms and Further Dependencies: Properties of Relational Decomposition, Algorithms for Relational Database Schema Design, Multi valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form. (12)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn Database Language – SQL
- Practice DDL, DML and DCL commands
- Demonstrate all built in arithmetic, aggregate, string, date and conversion functions
- Understand the concepts of functional dependencies
- Define Normalization and its normal forms

UNIT - V

Transaction Processing Concepts: Introduction to Transaction Processing, Transaction and System Concepts, Desirable properties of transactions, Characterizing Schedules based on recoverability, Characterizing Schedules based on serializability.
Concurrency Control Techniques and Database Recovery Techniques. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn introduction to transaction processing
- Know the transaction processing system and its properties
- Understand why concurrency is required
- Analyze the schedules based on recoverability and serializability
- Categorize the techniques for concurrency and database recove

Text Books:

1.Fundamentals of Database Systems by Ramez Elmasri and Shamkant B. Navathe, Pearson education, 5th edition, 2009.

Reference Books:

1. Database Concepts by Abraham Silberschatz, Henry F Korth, S.Sudarshan, McGraw,Hill, 6th edition, 2014.
2. An Introduction to Database Systems by C.J. Date, Addison Wesley, 8th edition, 2008.
3. Database Management Systems by Raghu Ramakrishnan, Johannes Gehrke, TMH, 2nd edition, 2000.

Course Outcomes:

- Able to understand the introductory concepts of databases and its architecture
- Draw E-R diagrams using conceptual modeling
- Develop relational algebra expressions
- Able to design a database using relational data model
- Able to Analyze the transaction processing, concurrency control and database recovery techniques

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A. – II SEMESTER
SCA 604 DATA STRUCTURES USING C++

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

Data Structure is a way of collecting and organizing data in such a way that can perform operations on these data in an effective way. It is about rendering data elements in terms of some relationship, for better organization and storage in different ways. This course will helps in understanding the various strategies require to solve a problem effectively and efficiently.

Course Objectives:

- To make the student understand the linear and non linear data structures
- To learn sorting and searching techniques and its efficiency.
- To Understand basic concepts about stacks, queues, lists, trees and graphs
- To understand about writing algorithms and step by step approach in solving problems with the help of fundamental data structures
- To apply the data structures and algorithms in real time applications and be able to design own data structure according to the application need.

UNIT -I

Fundamental Concepts: Introduction to Data Structures, Types of Data Structures, Relationship among data, data structures and algorithms, Implementation of data structures, Analysis of Algorithms, Complexity of algorithms: Space complexity, Time complexity Definition.

Linear Data Structure using Arrays: Sequential Organization, Linear Data Structure Using Sequential Organization: Array ADT, Memory Representation, Class Array, Multidimensional Arrays, Concept of Ordered List, Single Variable Polynomial: Representation, evaluation, Addition, Multiplication, Sparse Matrix: Representation, Addition, Transpose, String Manipulation Using Array, Pros and Cons of Arrays.

Searching: Search Techniques: Sequential search, Binary search, Fibonacci search, indexed sequential search, Hashed search.

Sorting: Types of sorting, General sort concepts, Bubble sort, Insertion sort, Selection sort, Quick sort, Radix sort, Merge sort, Heap sort. (15)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn the types of data structures
- Know the analysis of algorithms by learning the time and space complexities.
- Develop and run the linear data structures i.e., Arrays programs using C++
- Demonstrate the programs for polynomials, sparse matrix
- Experiment different searching and sorting techniques

UNIT - II

Stacks: Primitive operations, Stack Abstract Data Type, Representation of Stacks Using Sequential Organization, Multiple Stacks, Applications of Stacks – Expression Evaluation and Conversion, Processing of Function Calls.

Queues: Concept of Queues, Queue as Abstract Data Type, Realization of Queues Using Arrays, Circular Queue, Multi queues, Deque, Priority Queue, Applications of Queues: Job scheduling. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Understand the concepts Stack and its operations
- Experimenting the representation of stack using array and linked lists
- Demonstrate expression of evaluation and its conversion
- Understand the concepts Queue and its operations
- Learn the concepts of Queue, Circular queue, Deque and priority queue

UNIT -III

Linked Lists: Introduction, Linked List: Comparison of sequential and Linked Organizations, Terminology, Primitive operations, Realization of Linked Lists using arrays and dynamic memory management, Dynamic memory management in C++, Linked List Abstract Data Type, Linked List Variants, Doubly Linked List: Creation, Deletion, Insertion, Traversal, Circular Linked List, Linked Stack, Linked Queue. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Know the concepts of Linked Lists
- Comparison of Sequential and Linked Organization
- Understand the concepts of Dynamic memory management and its representation
- Demonstration of Linked list, Doubly linked lists creation, insertion and traversal in C++
- Develop Circular Linked list, Linked Stack and Linked Queue

UNIT - IV

Trees: Introduction, Basic terminology, General trees, Representation of a general tree, Types of Trees, Binary Tree, Properties, Binary Tree Abstract Data Type, Realization of a Binary Tree, Insertion of a Node in Binary Tree, Binary Tree Traversal (recursive traversals), Formation of binary tree from its traversals, Binary Search Tree: Inserting a node, Searching for a key, Deleting a node, Binary Tree and Binary Search Tree, Applications of Binary Trees: Expression tree, Decision tree, Huffman's coding.

Heaps: Basic Concepts, Implementation of Heap, Heap as Abstract Data Type. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn the basic terminology of general tree and its representation
- Understand the different types of trees
- Demonstrate binary tree insertion, traversals
- Experiment Binary Search Tree insertion, searching for a key
- Learn the basic concepts of Heap and its implementation

UNIT - V

More on Linked Lists: Copying a linked list, Computing the length of a linked list, Reversing singly linked list without temporary storage, Concatenating two linked lists, Erasing the linked list, Representation of Sparse Matrix Using Linked List, Application of Linked List.

Graphs: Introduction, Graph Abstract Data Type, Graph Representation, Graph Traversals, Spanning Trees: Prim's, Krushkal's Algorithm. (7)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn more concepts on Linked List
- Demonstrate how to concatenate two linked lists
- Experiment the representation of Sparse Matrix
- Understand the concepts of Graph, Graph traversals
- Develop applications of graphs in finding the shortest path

Text Books:

1.Data Structures using C++ by Varsha H.Patil, Oxford University Press, 2012.

Reference Books:

1.Data Structures Algorithms and Applications in C++ by Sartaj Sahani, University Press, 2nd Edition, 2011.

2. Data Structures Using C and C++ by Yedidyah Langsam, Moshe J Augenstein and Aaron M Tenenbaum, PHI, 2nd Edition, 2009.

3. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss, Pearson Education, 3rd edition, 2007.

4. Data Structures and Algorithms in C++ by Adam Drozdek, Cengage Learning, 4th Edition, 2013 .

Course Outcomes:

- Able to learn linear and nonlinear data structures
- Able to understand linear data structures
- Perform on different searching and sorting techniques
- Practice the concepts of stacks, queues and Linked lists
- Able to demonstrate on trees and graphs

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.-II SEMESTER
SCA 606 PROBABILITY AND STATISTICS

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Probability theory is important when it comes to evaluating statistics. Probability and Statistics for Computer Science treats the most common discrete and continuous distributions, showing how they find use in decision and estimation problems, and constructs computer algorithms for generating observations from the various distributions.

Course Objectives:

- To understand the difference between discrete and continuous random variables and probability
- To evaluate problems on discrete and continuous probability distributions
- To learn the basic concept and applications of correlation and regression
- Ability to implement various sampling techniques.
- To understand the concept of testing of hypothesis for large and small samples
- Ability to explore certain statistical concepts in practical applications of computer science areas.

UNIT - I

Probability: Sample Space, Events, Axiomatic Approach to Probability, Conditional Probability Independent Events, Baye's Formula with Applications.

Random Variables: Continuous and Discrete Random Variables, Distribution Function of a random variable, Expectation, Variance, Coefficient of Variation, Moment Generation Function. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- List the difference between discrete and continuous random variables
- Define sample space of an experiment and probability
- Evaluate the problems on probability and Baye's formula
- Evaluate distribution function of random variable
- Evaluate the problems on moment generating function

UNIT - II

Probability Distribution: Discrete Distributions, Binomial, Poisson and Geometric Distributions, Continuous Distributions, Uniform, Normal, Exponential. (10)

Learning Outcomes

By the end of this Unit, the student will be able to

- Compare Binomial and poisson distributions
- Illustrate the Geometric distribution
- Explain the difference between discrete and continuous probability distributions
- Apply normal distributions to various problems
- Compare normal distributions with other distributions

UNIT - III

Correlation and Regression: Correlation Coefficient, Rank Correlation Coefficient of Determination, Linear Regression, Methods of Least Squares, Fitting of the Curve of the Form ax + b, ax^2 + bx +c, ab^x , ax^b and ae^{bx} . (10)

Learning Outcomes

By the end of this Unit, the student will be able to

- To learn the basic concept and applications of correlation and regression
- Apply rank correlation coefficient to rank the quality
- Evaluate problems on correlation coefficient and rank correlation coefficient
- Explain the need of fitting of the curves
- Use of methods of least squares to fit a curve

UNIT - IV

Sampling Theory: Concepts of Sampling, Methods of Sampling, Simple Random Sampling, Systematic Sampling and Stratified Random Sampling (Descriptions Only), Concepts of Sampling Distributions and Standard Error, Point Estimation (Concepts only), Interval Estimation of Mean and Proportion. (10)

Learning Outcomes

By the end of this Unit, the student will be able to

- Explain the concept of sampling
- Summarize different methods of sampling
- Evaluate standard error
- Explain the concepts of sampling distributions
- Evaluate interval estimation of mean and proportion

UNIT - V

Test of Hypotheses: Critical Region, Two Types of Errors, Level of Significance, Large Sample Tests for Mean and Proportion, Exact Tests Based on t, F and Chi – Square Distributions. (10)

Learning Outcomes

By the end of this Unit, the student will be able to

- Compare two types of errors
- Explain the concept of critical region
- Test the hypothesis of large samples for mean
- Apply the concept of t distribution for small samples
- Explain the concepts of F and chi-square distributions

Text Book:

1. Fundamentals of Mathematical Statistics by S.C. Gupta & V.K. Kapoor, Sultan Chand & Sons, 2002.

Reference Books:

1**.** Probability and Statistics for Engineers by Irwin Millor and John E.Freund, PHI.

2. Probability and Statistics, Spiegel, TMH.

Course Outcomes:

- Able to understand the difference between discrete and continuous random variables and probability
- Define sample space of an experiment, Evaluate the problems on probability and Baye's formula
- Explain the difference between discrete and continuous probability distributions
- Summarize different methods of sampling
- Test the hypothesis of large samples for mean

Prepared by: Prof. N. Ravi Shankar Verified by : Prof. V. NagaLakshmi

MCA – II SEMESTER
SCA 608 OPERATING SYSTEMS

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

An operating system is an essential part of any computer system. This course will introduce the basic concepts of Operating system services, process management, memory management, device management, , file management and security & protection mechanisms. This course also will helps in learning the virtual memory concepts and synchronization.

Course Objectives:

- To know the evolution, types, structure and functions of operating systems
- To learn techniques involved in process, memory, device and file management
- To analyze the resource allocation
- To execute Linux basic commands and shell scripts
- To understand processor scheduling, synchronization, deadlocks and disk allocation algorithms
- To describe security and protection measures used in operating systems

UNIT - I

Introduction: Introduction to Operating Systems, Computer System Architecture, Operating System Structure, Operating System Operations, Distributed Systems, Special Purpose Systems, Computing Environments, Open-Source Operating Systems.

System Structures: Operating System Services, User Operating Systems, System Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation Operating System Structure, Virtual Machines. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn the system structure and operating system structure
- Understand operating system services
- List system calls and classify system calls
- Explain System programs
- Describe operating system structure and virtual machines

UNIT - II

Process Concept: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Communication in Client Server Systems.

Multithreaded Programming: Overview, Multithreading Models, Thread Libraries, Threading Issues, Operating System Examples.

Process Scheduling: Basic Concepts, Scheduling Criteria and Algorithms, Thread Scheduling, Multiple Processor Scheduling, Operating System Examples, Algorithm Evaluation. (12)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn process concept

- Understand process scheduling
- Describe interprocess communication
- Know multithreading
- Analyze scheduling algorithms

UNIT - III

Synchronization: Background, The Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors.

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock. (13)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define critical section problem
- Analyze Peterson's Solution
- Understand semaphores
- Learn the classic problems of Synchronization
- Determine Deadlock characterization

UNIT - IV

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation.

Virtual Memory Management: Background, Demand Paging, Copy on Write, Page Replacement, Allocation of Frames, Thrashing, Other Considerations. (13)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn memory allocation concepts
- Understand the concepts of paging, and the structure of page table
- Know Segmentation
- Interpret Demand paging
- Apply page replacement algorithms

UNIT - V

File System: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection.

Implementing File Systems: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, NFS, The WAFL File System. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn the file concept, access methods
- Understand the structure of directory and disk
- Know file sharing and protection
- Elaborate file system and directory implementation
- Discuss different allocation methods

Text Book:

1. Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Wiley Publications, 8th edition, 2012.

Reference Books:

1. Operating Systems by Achyut S. Godbole, Tata McGraw Hill, 3rd edition, 2010.
2. Operating Systems: Internals and Design Principles by William Stallings, Pearson Education, 7th edition, 2011.
3. Operating Systems: A Concept-based Approach by Dhamdhere, D.M., McGraw Hill, 2nd edition, 2006.

Course Outcomes:

- Understand the system structure and operating system structure
- Describe Operating system services, design and implementation
- Discuss the concepts of process, scheduling and multithreading
- Analyze critical section, semaphores and synchronization
- Determine Deadlock detection and prevention mechanisms
- Know file system and directory structures

Prepared by: Prof. V. NagaLakshmi

Verified by : Prof. V. NagaLakshmi

M.C.A.-II SEMESTER
SAE 604 TECHNICAL COMMUNICATION SKILLS

Hours per week: 4 End Examination: 60 Marks
Credits: 3 Sessionals: 40 Marks

Preamble:

Communication skills in the work front place an important role in shaping ones career. Therefore, professionals need to refine their communicative skills to be able to effectively interact with superiors, subordinates, peers, suppliers and customers successfully. The course has been designed to suit the purpose.

Course Objectives:

- To make the students learn the formation of sentences and improve the communication skills
- To familiarize the students with the use of English in communication and help them to acquire the ability to expand their linguistic resources to enhance communicative competence.
- To develop listening, speaking, reading and writing skills in such a way that they improve the ability to exchange information, grasp and express ideas, feelings etc., with clarity and confidence.
- To make them interview ready and to increase the employability of the students by improving their overall communicative efficiency.

UNIT - I

Features of Indian English: Correction of sentences, Structures, Tenses, ambiguity, idiomatic distortions and misappropriations. (8)

Learning Outcomes:

By the end of the unit, the student will be able to:

- Understand the features of Indian English
- Identify incorrect sentences in English and write their correct form
- Use structures and tenses accurately
- Point out distorted idiomatic expressions and use their write form

UNIT - II

Informal conversation Vs Formal expression: Features of good communication, Different flows of communication, Verbal and Nonverbal communication, Barriers to effective communication –ways to overcome the barriers. (10)

Learning Outcomes:

By the end of the unit, the student will be able to:

- Comprehend the characteristics of good communication
- Figure out the different flows of communication in an organization and their uses
- Understand the importance of verbal communication and use oral and written communication effectively
- Relate the significance of non-verbal communication in the entire communication process
- Become aware of the obstacles that hinder effective communication to take place and the different strategies to overcome them

UNIT - III

Types of Communication: Oral, aural, Writing and reading, Word-Power, Vocabulary, Jargon, rate of speech, pitch, tone, Clarity of voice, Group discussion, Personality traits, types of group discussion, Team player, Leadership qualities. (12)

Learning Outcomes:

By the end of the unit, the student will be able to:

- Learn and appreciate the different types of communication and their uses
- Amass strong comprehensive vocabulary that is useful for specific purposes
- integrate the techniques of speech such as pitch, intonation, clarity and rate of speech
- Prioritize the personality traits and skills required to effectively participate in a G.D

UNIT - IV

Formal and informal interviews: Ambiance and polemics, interviewing in different settings and for different purposes e.g., Eliciting and giving information, Preparation for a job interview, Personality traits assessment, Recruiting, Performance appraisal. (12)

Learning Outcomes:

By the end of the unit, the student will be able to:

- Analyze the purpose of interviews
- Assess the processes involved in different types of interviews
- Plan how to prepare for an interview
- Prepare how to answer common interview questions

UNIT - V

Technical presentations: Types of presentation, Video conferencing, Participation in meetings, chairing sessions.

Letter-writing, business letters, Proforma, Format, Style, Effectiveness, Promptness, Analysis of sample letters collected from industry, email. (12)

Learning Outcomes:

At the end of the unit, the student will be able to:

- Overcome nervousness prior to any presentation
- Prepare and structure good presentations that deliver effective messages
- Facilitate and participate in a video conference effectively
- Evaluate the roles and responsibilities of a member in the meeting
- Produce skills in leadership, time management, decision making and communicate effectively with members in a meeting
- Write a business letter, which includes appropriate greetings, heading, closing and a professional tone

Text Books:

1. Essentials of Business Communication by Rajendra Pal & J S KorlahaHi, Sultan Chand & Sons.
2. Advanced Communication Skills by V. Prasad, Atma Ram Publications.

Course Outcomes:

- Able to learn the formation of sentences and improve the communication skills
- Understand the importance of verbal communication and use oral and written communication effectively
- Integrate the techniques of speech such as pitch, intonation, clarity and rate of speech
- Produce skills in leadership, time management, decision making and communicate effectively with members in a meeting
- Interview ready and to increase the employability of the students by improving their overall communicative efficiency.

Prepared by: Dr. Sushma Raj Verified by : Prof. V. NagaLakshmi

M.C.A.-II SEMESTER

SCA 622 DATABASE MANAGEMENT SYSTEM LAB

Hours per week: 4 Examination: 100 Marks

Credits: 2

1. Creation, altering and dropping of tables using SQL.
2. Implementing integrity Constraints on tables.
3. Implementing DML Operations using SQL- Insert, Delete, Update
4. Simple Queries to access data from Tables using Select Statement and where condition using Distinct, And, Or, Not And Order By Operators.
5. i) Queries Using Built in Functions :
   Arithmetic Functions: Sign, Abs, Ceil, Floor, Exp, Power, Log, Sqrt,
   String Functions: Concat, Lpad, Rpad, Ltrim, Rtrim, Lower, Upper, Initcap, Length, Substr and Instr.
   Date Functions: Sysdate, Next_Day, Add_Months, Last_Day, Months_Between, Least, Greatest, Trunc, Round.
   Aggregate Functions: Count, Sum, Avg, Max And Min, Group by, Having,
   ii) Queries Using Conversion Functions: To_Char, To_Number and To_Date
   iii) Queries Using Set Operators: Union, Intersect, Minus
6. i) Queries Using Joins, Natural Join, Innerjoin, Outer Join.
   ii) Queries Along with Sub Queries and Correlated Queries using Any, All, In, Exists, Notexists.
7. i) Creating Other Schema Objects**:** Defining Views, Creating Views, using Views to Change Data, Dropping Views, Creating Indexes and Sequences.
   ii) Using DCL Commands: Commit and Rollback.
8. Creation of Simple PL/SQL Program which includes Declaration Section, Executable Section, Select ... into Clause
9. Develop Programs that include Features of Nested If and Case.
10. Program Development using While Loop, For Loop, Nested Loops

Reference Books:

1. SQL, PL/SQL The programming language of ORACLE by Ivan bayross, BPB publications, 4th edition, 2009.
2. Programming Oracle triggers and Stored Procedures by Kevin Owens, PHI, 3rd Edition, 2003.

Course outcomes:

- Able to learn introduction to Database Language, DDL, DML, DCL
- Learn how to create database and how to perform alteration to the table structure using ALTER TABLE
- Understands the concept of RDBMS through entity integrity(primary key) and referential integrity(foreign key)
- Experiments to insert, retrieve, update and delete from the table data
- Performs all arithmetic, string, date and aggregate functions using database language
- Construct simple queries, subqueries and complex queries
- Able to create other schema objects like views, indexes and sequence
- Able to develop PL/SQL programs

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.-II SEMESTER

SCA624 DATA STRUCTURES USING C++ LAB

Hours per week: 4 Examination: 100 Marks

Credits: 2

1. Implementation of Linear Search, Binary Search.
2. Implementing the following sorting methods.
    a. Bubble sort
    b. Insertion sort
    c. Selection Sort
    d. Quick Sort
    e. Merge Sort
3. Implementation of Polynomial Addition using Arrays.
4. Implementation of Sparse Matrix addition and Multiplication using Arrays.
5. Array implementation of stack.
6. Array implementation of Queue.
7. Implementation to convert infix expressions to post fix notation, prefix notation.
8. Simple expression evaluator that can handle +,-, /, *.
9. Singly Linked List operations – insertion, deletion, display, reverse.
10. Implementation of Linked Stack Operations.
11. Implementation of Linked Queue Operations.
12. Implementation of circular queue ADT using an array.
13. Implementation of Doubly Linked List.
14. Implementation of Circular Linked List.
15. Implementation of Binary Tree, Binary Search Tree creation, traversals.
16. Implementation of Binary Search Tree operations, insertion all cases and deletion all cases.
17. Implementation of Graph Traversals.

Reference Books:

1. Data Structures through C++ by Varsha H Patil, Oxford University Press, New Edition, 2011.
2. Data Structures through C in depth by S.K. Srivastva and Deepali Srivastva, BPB publications, 2004.
3. Data Structures and Algorithms in C++ by Adam Drozdek, Cengage Learning, 4th edition, 2013.

Course Outcomes:

- Able to perform searching techniques
- Able to differentiate between different sorting techniques
- Understand Linear and Non-linear Data structure
- Implement the operations on Stacks, Queues using Arrays
- Experiments on conversions of expression and expression evaluation by using prefix, postfix and infix
- Able to develop programs on operations of stacks, queues, circular queues, using linked lists
- Able to demonstrate on binary trees and operations on binary search trees and tree traversals

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A-II SEMESTER
SCA 626 PROGRAMMING WITH UNIX LAB

Hours per week: 4 Examination: 100 Marks

Credits: 2

1. **Unix Utilities** – Introduction to Unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, cp, mv ln, rm, unlink, mkdir, rmdir, du, df, mount, umount find, unmask, ulimit, ps, who, finger, arp, ftp, telnet, rlogin, text processing utilities and backup utilities, detailed commands to be covered are cat, tail, head, sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm., cmp, diff, tr, awk, tar, cpio.

2. Study about the General Purpose Utilities.

a) Banner b) cal c) date d) calendar e) tty f) bc g) spell & fspell

3. Write a shell program to find the Sum of digits in a 3 digit number using while loop.

4. Write a shell program to print first 'n' terms of Fibonacci series.

Programs on Processes:

5. Chain of processes.

6. Fan of processes.

Programs on Process Scheduling:

7. FCFS scheduling algorithm

8. Round Robin scheduling algorithm

9. Priority scheduling algorithm (Preemptive, Non Preemptive)

10. Shortest job First scheduling algorithm (Preemptive, Non Preemptive)

Reference Books:

1. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill, 4th edition, 2006.

2. Unix networking program by Stevens W. Richard, 2005.

3. Advanced Unix programming by H.J.Rechkind, Pearson Education, 2nd edition, 2004.

Course Outcomes:

- Able to learn UNIX utilities and practice UNIX commands
- Able to understand how to work in Vi Editor environment and run Shell programming
- Develop Operating systems programmes like chain of processes and fan of processes
- Design scheduling algorithms

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.-III SEMESTER
SCA701 PROGRAMMING WITH JAVA

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

The aim of the course is to make the students learn the basic concepts of Java programming. This course covers preliminaries and makes the students learn how to program in java using Basic Concepts, Inheritance, Interfaces, Packages, Threads, I/Os, Applets, Swings, Event Handling, Collections and allow the students to implement effectively.

Course Objectives:

- To introduce the object oriented programming concepts.
- To understand object oriented programming concepts, and apply them in solving problems.
- To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
- To introduce the implementation of packages and interfaces
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using applets and swing controls.

UNIT - I

Object-oriented thinking- A way of viewing world – Agents and Communities, messages and methods, Responsibilities, Classes and Instances, Class Hierarchies-Inheritance, Method binding, Overriding and Exceptions, Summary of Object-Oriented concepts. Java buzzwords, An Overview of Java, Data types, Variables and Arrays, operators, expressions, control statements, Introducing classes, Methods and Classes, String handling.

Inheritance– Inheritance concept, Inheritance basics, Member access, Constructors, Creating Multilevel hierarchy, super uses, using final with inheritance, Polymorphism-ad hoc polymorphism, pure polymorphism, method overriding, abstract classes, Object class, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance.

Learning Outcomes:

By the end of the unit the student will be able to

- Learn the OOP techniques.
- Understands he concepts of Classes and Instances.
- Practice the concepts of Inheritance, Polymorphism and method overriding.
- Demonstrate the concepts of Constructors, Creating Multilevel hierarchy, and super uses.

UNIT - II

Packages- Defining a Package, CLASSPATH, Access protection, importing packages.

Interfaces- defining an interface, implementing interfaces, Nested interfaces, applying interfaces, variables in interfaces and extending interfaces.

Stream based I/O(java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files,

Random access file operations, The Console class, Serialization, Enumerations, auto boxing, generics.

Learning Outcomes:

By the end of the unit the student will be able to

- Understand the concepts of Package, CLASSPATH, Access protection.
- Design interfaces, Nested interfaces.
- Demonstrate File class, Reading and writing Files
- Illustrate auto boxing and generics.

UNIT - III

Exception handling - Fundamentals of exception handling, Exception types, Termination or resumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built- in exceptions, creating own exception sub classes.

Multithreading- Differences between thread-based multitasking and process-based multitasking, Java thread model, creating threads, thread priorities, synchronizing threads, inter thread communication.

Learning Outcomes:

By the end of the unit the student will be able to

- Learn the concepts of Exception handling and Exception types.
- Understand the concepts of throw, throws and built- in exceptions.
- Differentiate process-based multitasking and Java thread model
- Summarize synchronizing threads and inter thread communication.

UNIT - IV

The Collections Framework (java.util)- Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hashtable , Properties, Stack, Vector

More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner

Learning Outcomes:

By the end of the unit the student will be able to

- Learn the concepts of Collection Interfaces.
- Understand Collection classes- Array List, Linked List and Hash Set.
- Explain the concepts of Map Interfaces and Classes and Comparators
- Discuss about Utility classes, String Tokenizer, Bit Set, Date and Calendar.

UNIT - V

GUI Programming with Swing – Introduction, limitations of AWT, MVC architecture, components, containers. Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

A Simple Swing Application, **Applets** – Applets and HTML, Security Issues, Applets and Applications, passing parameters to applets. Creating a Swing Applet, Painting in

Swing, A Paint example, Exploring Swing Controls- JLabel and Image Icon, JText Field, The Swing Buttons- JButton, JToggle Button, JCheck Box, JRadio Button, JTabbed Pane, JScroll Pane, JList, JCombo Box, Swing Menus, Dialogs.

Learning Outcomes:

By the end of the unit the student will be able to

- Learn the concepts of MVC architecture, components, containers.
- Understand Layout Managers
- Design Flow Layout, Border Layout, Grid Layout, Card Layout and Grid Bag Layout
- Compare Adapter classes, Inner classes and Anonymous Inner classes.
- Illustrate Applet and Swing Controls

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

Reference Books:

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons.
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Object Oriented Programming through Java, P. Radha Krishna, Universities Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.
5. Java Programming and Object oriented Application Development, R. A. Johnson, Cengage Learning.

Course outcomes:

- Understand the need of object oriented programming using Java programming language.
- Able to define classes, invoking methods, class libraries auto boxing and generics.
- Able to learn the concepts of exception handling and Exception types.
- Identify the concepts of synchronizing threads and inter thread communication.
- Able to define the concepts of Collection Interfaces.
- Understand the concepts of Collection classes- Array List, Linked List and Hash Set
- Identify concepts of Map Interfaces and Classes and Comparators.
- Able to understand the need of the concepts of – Applets, and Swing Controls.

Prepared by: Mr. G. Babu Rao Verified by : Prof. V. NagaLakshmi

M.C.A.-III SEMESTER
SCA703 DESIGN AND ANALYSIS OF ALGORITHMS

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

This core course covers good principles of algorithm design, elementary analysis of algorithms, and fundamental data structures. The emphasis is on choosing appropriate data structures and designing correct and efficient algorithms to operate on these data structures.

Course Objectives:

- To introduce students, the concepts of algorithm analysis to find out the space and time complexity of different algorithms.
- Different design techniques such as greedy method, divide and conquer, backtracking, dynamic programming, branch and bound are to be studied for finding the solution to the different problems.
- Ability to solve problems independently and think critically and to understand, estimate the performance of algorithm.

UNIT – I

Introduction to Algorithms: Algorithm Specification, Performance Analysis, Introduction to Randomized Algorithms. Divide And Conquer: The General Method, Binary Search, Finding Maximum and Minimum, Quick Sort, Selection, Strassen„s Matrix Multiplication. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Know the concepts of space and time complexity of different algorithms
- Understand Divide And Conquer methods
- Find maximum and minimum paths using General Method, Binary Search,
- Demonstrate on Quick Sort, Selection Sort and Strassen's Matrix Multiplication.

UNIT - II

The Greedy Method: The General Method, Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum Cost Spanning Trees, Single Source Shortest Paths. (7)

Learning Outcomes:

By the end of the unit the student will be able to

- Define Knapsack Problem and Tree Vertex Splitting
- Understand Job Sequencing with Deadlines
- Learn Minimum Cost Spanning Trees, Single Source Shortest Paths

UNIT – III

Dynamic Programming: The General Method, Multistage Graphs, All Pairs Shortest Paths, Optimal Binary Search Trees, String Editing, Reliability Design, The Traveling Sales Person Problem. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Compare Multistage Graphs, and All Pairs Shortest Paths
- Design Optimal Binary Search Trees
- Understand String Editing and Reliability Design
- Illustrate The Traveling Sales Person Problem

UNIT – IV

Basic Traversal and Search Techniques: Techniques for Graphs, Connected Components and Spanning Trees, Bi-connected Components and DFS. Back Tracking: The General Method, Eight Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Learn Techniques for Graphs, Connected Components and Spanning Trees
- List Bi-connected Components and DFS
- Understand The General Method and Eight Queens Problem
- Illustrate Sum of Subsets, Graph Coloring and Hamiltonian Cycles

UNIT - V

Branch And Bound: The Method, 0/1 Knapsack Problem, Traveling Salesperson Problem Algebraic Problems: The General Method, Evaluation and Interpolation. NP Hard and NP Complete Problems: Basic Concepts. (8)

Learning Outcomes:

By the end of the unit the student will be able to

- Explain the concepts of The Method, 0/1 Knapsack Problem.
- Explain the concepts of Traveling Salesperson Problem Algebraic Problems.
- Explain the concepts of The General Method, Evaluation and Interpolation
- Explain the concepts of NP Hard and NP Complete Problems

Text Book:

1. Fundamentals of Computer Algorithms by Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, University Press, 2nd Edition, 2008.

Reference Books:

1. Fundamentals of Algorithmics by G. Brassard and P.Bratley, PHI, 2011.
2. Introduction to Algorithms by T.H. Cormen, C.E. Leiserson, R.L.Rivest, 3rd edition, PHI, 2010 .
3. Introduction to Design Analysis of Algorithms by Anany Levitin, 2nd edition, Pearson Publications, 2009.

Course Outcomes:

- Understand the concepts of Worst case and average case analysis. Asymptotics and "big O" notation. Polynomial and exponential growth.
- Understand the concepts of Knapsack Problem, Tree Vertex Splitting.
- Able to compare Multistage Graphs, All Pairs Shortest Paths
- Able to understand the concepts Bi-connected Components and DFS.
- Able to understand the concepts of The General Method, Eight Queens Problem
- Find the need of NP Hard and NP Complete Problems

Prepared by: Mr. G. Babu Rao Verified by : Prof. V. NagaLakshmi

M.C.A. - III SEMESTER
SCA 705 OPTIMIZATION TECHNIQUES

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

The optimization techniques are useful in finding the optimum solution or unconstrained maxima or minima of continuous and differentiable functions. This course have had successful applications in business and public services.

Course Objectives:

- To learn the basic concepts of Operations Research models and techniques
- To understand the mathematical formulation of business problem and to solve the linear programming problems.
- To learn the applications of transportation problems and to know the methods of solving transportation and assignment problems.
- Ability to know the concept of game theory and game strategies and also the algorithms related to it.
- To understand the concept of job sequencing and to know the method of solving sequencing problems.
- To learn how to draw the project network diagram for any project description and also to solve project network problems.

UNIT - I

Overview of Operations Research: OR Models, OR Techniques.

Linear Programming: Introduction, Mathematical Formulation, Graphical Solution, Basic Feasible Solutions, Simplex Algorithm, Artificial Variables, Big M and Two Phase Method, Degeneracy, Alternative Optima, Unbounded Solutions, Infeasible Solutions. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define the mathematical formulation of given business problem.
- Apply graphical method to solve linear programming problem.
- Maximize or Minimize the objective function of a Linear programming problem for the given constraints using simplex method.
- Maximize or Minimize the objective function of a Linear programming problem using artificial variable technique.
- Maximize or Minimize the objective function of a Linear programming problem using Big M and Two phase method.

UNIT - II

Dual in Linear Programming: Relation between Primal and Dual problems, Dual Simplex Method.

Integer Programming: Gomory's Cutting Plane Algorithm, Branch and Bound Algorithm (8)

Learning Outcomes

By the end of this Unit, the student will be able to

- Compare primal and dual problems.
- Maximize or Minimize the objective function of a Linear programming problem using dual simplex method.

- Maximize or Minimize the objective function of a Linear programming problem using integer programming algorithms.

UNIT - III

Transportation Problems: Initial Basic Feasible Solutions-North West Corner Rule, Lowest Cost Entry Method, Vogels Approximation Method, Optimal Solution-MODI Method,
Assignment Problems: Hungarian Method, Travelling Salesman Problem. (10)

Learning Outcomes

By the end of this Unit, the student will be able to

- Model transportation problem for a business problem
- Illustrate the initial basic feasible solutions using various methods
- Evaluate optimal solution of a transportation problem using initial basic feasible solution
- Evaluate assignment problems using Hungarian method.
- Evaluate travelling salesman problem

UNIT - IV

Game Theory: Two Person Zero Sum Games, Mixed Strategy Games and their Algorithms. (8)

Learning Outcomes

By the end of this Unit, the student will be able to

- Explain the game theory and game strategies
- Evaluate two person zero sum games
- Evaluate mixed strategy games
- Explain the algorithms about games

UNIT - V

Job Sequencing: Introduction, Solution of Sequencing Problems, Processing of n Jobs through two Machines, n Jobs through 3 Machines and n Jobs through m Machines.
Network Models: Definitions, CPM and PERT their algorithms. (8)

Learning Outcomes

By the end of this Unit, the student will be able to

- Illustrate the method of job sequencing
- Evaluate sequencing problems
- Evaluate processing of n jobs through two machines
- Demonstrate how to draw project network diagrams for any project description
- Evaluate earliest times, latest times and critical path of a project network using CPM and PERT

Text Book:

1. Operations Research by Sharma S.D. Keder Nath, Ram Nath & Co.

Reference Books:

1. Operations Research – An Introduction by Handy A. Taha, Pearson Education, 8th Edition, 2008
2. Operations Research by Kanti Swaroop, Manmohan and P.K. Gupta, Sultan Chand & Sons, 4th edition.

Course Outcomes:

- Learn the basic concepts of Operations Research models and techniques
- Apply graphical method to solve linear programming problem.
- Able to perform problems to maximize or minimize the objective function of a Linear programming problem for the given constraints using simplex method, using artificial variable technique, and using Big M and Two phase method
- Practice on maximize or minimize the objective function of a Linear programming problem using dual simplex method and using integer programming algorithms
- Evaluate optimal solution of a transportation problem, assignment problem and travelling salesman problem
- Illustrate the method of job sequencing

Prepared by: Prof. N. Ravi Shankar Verified by : Prof. V. NagaLakshmi

M.C.A.-III SEMESTER

SCA 707 SOFTWARE ENGINEERING

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40

Preamble:

This course will emphasize on the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation testing, and documentation of software.

Course Objectives:

- The objective of this course is to introduce various software engineering concepts and principles.
- To introduce the study of various phases of software engineering.
- To understand various process models and requirement engineering
- To develop design engineering and various testing strategies & techniques.

UNIT – I

Software and Software Engineering - The Nature of Software, The Unique Nature of Web Apps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths, How it all Starts. Process Models: A Generic Process Model, Process assessment and improvement, Prescriptive process models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Technology, Product and Process. (12)

Learning Outcomes:

By the end of the unit the student will be able to

- Define the Nature of Software, The Unique Nature of Web Apps, Software Engineering
- Extend the concepts of the Software Process, Software Engineering Practice
- Illustrate Unified Process, Personal and Team Process Models
- Summarize Process Technology, Product and Process

UNIT – II

Understanding Requirements: Requirements Engineering, Establishing the groundwork, Eliciting Requirements. Developing Use Cases, Building requirements model, Negotiating requirements, Validating Requirements. Requirements Modeling - Scenarios, Information And Analysis Classes: Requirement Analysis, Scenario – Based Modeling, UML Models That Supplements the Use Case, Data Modeling Concepts, Class Based Modeling. (12)

Learning Outcomes:

By the end of the unit the student will be able to

- Explain the concepts of Requirements Engineering, Eliciting Requirements
- Develop Use Cases, Building requirements model
- Illustrate Negotiating requirements, Validating Requirements. Requirements Modeling

UNIT - III

Design Concepts: Design within the Context of Software Engineering, The Design Process, Design Concepts, The Design Model. Architectural Design: Software Architecture, Architectural Genres, Architectural Styles, Architectural Design, Assessing Alternative Architectural Design, Architectural Mapping Data Flow. (10)

Learning Outcomes:

By the end of the unit the student will be able to

- Define Design Process, Design Concepts and the Design Model.
- Understand Architectural Design Software Architecture, Architectural Genres
- Explain the concepts of Architectural Mapping Data Flow.

UNIT – IV

Component level Design: What is Component, Designing Class Based Components, Conducting Component Level Design, Component-Level Design for Web Apps, Designing Traditional components, Component Based Development. User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps. (12)

Learning Outcomes:

By the end of the unit the student will be able to

- List Designing Class Based Components,
- Develop Component-Level Design for Web Apps
- Illustrate the Golden Rules, User Interface Analysis and Design,
- Understand Interface Analysis, Interface Design Steps

UNIT-V

Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Validation Testing, System Testing, The Art of Debugging. Testing Conventional Applications: Software Testing Fundamentals, Internal and External View of Testing, White Box Testing, Basis Path Testing, Control Structure Testing, Black Box Testing. (12)

Learning Outcomes:

By the end of the unit the student will be able to

- Define Test Strategies for Conventional Software
- Understands the concepts of Software Testing Fundamentals, Internal and External View of Testing
- Know White Box Testing and Basis Path Testing
- Explain the concepts of Control Structure Testing, Black Box Testing.

Text Book:

1. Software Engineering- A Practitioners Approach by Roger S.Pressman, McGraw Hill Edition, 7th edition, 2010.

Course Outcomes:

- Understand the need of The Software Process, Software Engineering Practice
- Able to define The Unified Process, Personal and Team Process Models
- Identify the concepts of Requirements Engineering, Eliciting Requirements
- Able to Develop Use Cases, Building requirements model.
- Identify the concepts of Architectural Design: Software Architecture, Architectural Genres and Architectural Mapping Data Flow
- Identify Test Strategies for Conventional Software, Validation

Prepared by: Mr. G. Babu Rao Verified by : Prof. V. NagaLakshmi

M.C.A.-III SEMESTER
SOE XXX 721 OPEN ELECTIVE

Hours per week: 3 | End Examination: 60 Marks

Credits: 3 | Sessionals: 40

M.C.A.-III SEMESTER
SCA721 PROGRAMMING WITH JAVA LAB

Hours per week: 3 Examination: 100 Marks

Credits: 2

I. Use Linux and MySQL for the Lab Experiments. Though not mandatory, encourage the use of Eclipse platform.

II. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed.

1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
3. A) Develop an applet in Java that displays a simple message.
   b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
5. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
6. Write a Java program for the following:
   i) Create a doubly linked list of elements.
   ii) Delete a given element from the above list.
   iii) Display the contents of the list after deletion.
7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.
8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.
10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
12. Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.
13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order
15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

Reference Books:

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

Course Outcomes:

- Able to write programs for solving real world problems using java collection frame work.
- Able to write programs using abstract classes.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java.
- Able to demonstrate on files

Prepared by: Mr. G. Babu Rao Verified by : Prof. V. NagaLakshmi

M.C.A.- III SEMESTER
SCA 723 MEAN STACK LAB

Hours per week : 4 Examination: 100 Marks

Credits: 2

1. Introducing the Node.js
2. Angular stack
    i) Understanding the basic web development framework
    ii) Understanding the node.js to Angular Stack Components
3. Understanding Java Script
4. Getting Started with Node.js
5. Creating a Node.js application
6. Implementing http services in Node.js
7. Implementing socket services in Node.js
8. Building MongoDB environment and managing collection
9. Manipulating MongoDB documents from Node.js
10. Accessing MongoDB from Node.js
11. Implementing Express in Node.js
12. Understanding Angular and Creating a basic Angular application

Reference Books:

1.Node.js, MongoDB and Angular WebDevelopment by Brad Dayley, Brendan Dayley, CalebDayley, Addison Wesley publications, 2nd Edition.

2.Full Stack Javascript Development with MEAN by Adam Bretz & Colin Jhrig, sitepoint publication, Cambridge street, Australia, 2017.

Course Out comes:

- Learn Node.js
- Practices web development framework through Angular stack
- Understands Java Script
- Creates Node.js application
- Builds Mongo DB application
- Implements Express in Node.js
- Develops a basic Angular application

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A. III SEMESTER
SCA 791 MINOR PROJECT

Credits: 2 Examination: 100 Marks

M.C.A.-IV SEMESTER
SCA 702 COMPUTER NETWORKS

Hours per week: 4 End Examination: 60 Marks

Credits 4 Sessionals: 40 Marks

Preamble:

The course is designed to impart a basic understanding of the working of computer networks, with the Internet as the case in point. Starting with the application layer with which the user interacts directly, it covers the important principles and protocols in the application, transport, network and link layers. Brief introduction to socket programming and wireless networks are given.

Course Objectives:

- To make the students understand the division of network functionalities into layers
- To be familiar with the components required to build different types of networks
- To learn the functionalities of each layers
- To develop simple network applications using socket programming
- To understand the working of TCP and UDP
- To know the protocols used for multiple access links

UNIT - I

Introduction: Uses of the computer networks, Network Haradware, References models, network standardization.

The Physical Layer: Guided Transmission Media, Wireless Transmission,

The Data Link Layer: Data Link Layer Design Issues, Error Detection and Correction, Sliding window protocol.

Learning Outcomes:

After completion of this unit, student will be able to

- Identify the roles of the various components of the Internet
- Explain network parameters such as delay, loss and throughput
- Model the network using a layered architecture
- Learn error detection and correction codes

UNIT - II

Medium Access Control Sublayer, Multiple access protocols, Ethernet, Wireless LANs, Data Link Layer Switching.

Learning Outcomes:

By the end of the unit the student will be able to

- Summarize the principles governing the working of network applications
- Outline the working of popular applications in the Internet
- Understand data link layer switching
- Develop simple network applications using socket programming

UNIT - III

Network Layer: Network layer design issues, Routing algorithms, Internetworking, The Network Layer in the Internet.

Learning Outcomes:

By the end of the unit the student will be able to

- Distinguish between virtual circuit and datagram networks
- Outline the working of the Internet Protocol
- Understand the different types of addresses
- Explain and analyze the working of routing algorithms

UNIT - IV

Transport Layer: The Transport service, Elements of transport protocols. The Internet Transport Protocols: UDP, TCP.

Learning Outcomes:

By the end of the unit the student will be able to

- Explain the need for multiplexing and demultiplexing at the transport layer
- Compare connectionless service with connection-oriented service
- Outline the working of TCP and UDP
- Analyze the principles of congestion control

UNIT - V

Application Layer: The Domain Name System, Electronic Mail, World Wide Web.

Learning Outcomes:

By the end of the unit the student will be able to

- Understand domain name system
- Summarize the protocols used for multiple access links
- Compare the characteristics of wireless networks with those of wired networks
- Outline the working of IEEE 802.11 standard and Mobile IP

Text Book:

1. Computer Networks by Andrew S Tanenbaum and David J Wetherall, Pearson education, 5th edition, 2011.

Reference Books:

1. An Engineering Approach to Computer Networks by S.Keshav, Pearson Education, , 2nd edition .
2. Computer Networks: A Systems Approach by Larry L Peterson and Bruce S Davi, Elsevier Publication, 4th Edition, 2003.
3. Computer Networks: A TopDown Approach by Behrouz A Forouzan and Firouz Mosharraf, TMH Publications, 2011.

Course Outcomes:

- Able to interpret the concept of modular network design using layered protocol architecture
- List the various components in the Internet and their functions
- Able to analyze various types of services provided by each layer in the network architecture
- Discuss the working of the important protocols used in the Internet
- Develop simple network applications and test them

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A.- IV SEMESTER
SCA 704 DATA MINING

Hours per week: 4 End Examination: 60 Marks

Credits 4 Sessionals: 40 Marks

Preamble:

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. This course covers supervised, un supervised learning techniques and also anomaly detecting techniques.

Course Objectives:

- To understand the basic concepts of knowledge discovery from databases.
- To learn various classification techniques and its implementations.
- To understand about alternative techniques of classification.
- To make the student understand the association rules and frequent item sets.
- To give a overview to the student regarding various cluster techniques.
- To make the student to understand how anomaly detection is done.

UNIT - I

Introduction: What is Data mining? Motivating Challenges, The origins of Data mining, Data Mining tasks.

Data: Types of Data, Data Quality, Data Preprocessing, Measures of Similarity and Dissimilarity. (8)

Learning Outcomes:

By the end of this unit, the student will be able to

- Define Data mining
- Identify the challenges in data mining
- Analyze various data mining tasks
- Illustrate the steps in data preprocessing
- Compare the measures of similarity ad dissimilarity

UNIT - II

Classification: Basic Concepts, General framework for Classifier, Decision Tree Classifier, Model Over fitting, Model Selection, Model Evaluation, Presence of Hyper parameters, Pit falls of Model selection and Evaluation, Model Comparison. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- How a decision tree classifier works?
- Illustrate how model over fitting is done
- Identify model evaluation techniques
- Methods of comparing classifiers
- Explain the various model selection and evaluation techniques

UNIT - III

Classification: Alternative Techniques-Types of Classifier, Rule-Based Classifier, Nearest-Neighbor classifiers, Naïve Bayes Classifier, Bayesian Networks, Logistic Regression Classifier, Support Vector Machine (SVM), Ensemble Methods, Class Imbalance Problem.

Association Analysis: Basic Concepts and Algorithms-Problem Definition, Frequent Item sets Generation, Rule Generation, Compact Representation of Frequent Item sets, Alternative Methods for Generating Frequent Item sets, FP-Growth Algorithm. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- Illustrating the characteristics of rule based classifier
- Explaining about Support Vector Machine
- Identifying the frequent item sets, using frequent item set generation rule.
- Listing the alternative methods of generating frequent item sets
- Constructing an FP growth algorithm

UNIT - IV

Cluster Analysis: Basic Concepts and Algorithms - Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation.

Cluster Analysis: Additional Issues and Algorithms - Characteristics of Data, Cluster, and Clustering Algorithms, Prototype-Based Clustering, Density-Based Clustering, Graph-based Clustering, Scalable Clustering Algorithms. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define K-Means algorithm
- Illustrate various clustering algorithms
- Classify various Prototype-Based Clustering
- Construct Graph Based Clustering And Scalable Clustering Algorithm
- Compare which algorithm is best

UNIT - V

Anomaly Detection: Characteristics of Anomaly Detection Problem, Characteristics of Anomaly Detection Methods, Statistical Approaches, Proximity-Based Approaches, Clustering Based Approaches, Reconstruction Based approaches. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- List the characteristics of Anomaly Detection Methods
- Explain the Statistical approaches
- Understand the concepts of Clustering Based Approaches
- Understand the concepts of Proximity Based Approaches
- Understand the concept of Various Reconstruction Based Approaches

Text Book:

1. Introduction to Data Mining by Pang- Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education, Second Edition.

Reference Books:

1. Data Mining Concepts and Techniques by Jiawei Han, Micheline Kamber, and Jian Pei, Elsevier Publications,3rd Edition**,** 2013.
2. Insight to Data Mining Theory and Practice by K.P.Soman, Shyam Diwakar and V. Ajay, Prentice Hall of India, 2006.

Course outcomes:

- Able to learn introductory concepts of datamining
- Able to understand classification by model selection, evaluation and comparison
- Explains the other classification techniques by using Naïve's classifier, Baye's classification
- Able to practices different association analysis algorithms
- Able to illustrate various clustering algorithms

Prepared by: Dr. M. Sesha Shyaee Verified by : Prof. V. NagaLakshmi

M.C.A. - IV SEMESTER
SCA 706 ARTIFICIAL INTELLIGENCE

Hours per week: 4 End Examination: 60 Marks
Credits 4 Sessionals : 40 Marks

Preamble:

This course enables the students to think critically about what makes humans intelligent, and how computer scientists are designing computers to act more like us. Artificial Intelligence (AI) is the study of how to make computers make things which at the moment people do better. AI plays an important role in the design and development of systems with intelligent behavior.

Course Objectives:

- To impart knowledge on the fundamentals of Artificial Intelligence,
- To learn the concept of Intelligent Agents and problem solving process through uninformed and informed searches.
- To know how to gain an insight into competitive environments which give rise to adversarial search problems, often known as games.
- To view many problems in AI as problems of constraint satisfaction.
- To get complete idea of knowledge representation and understand the two knowledge representation techniques Propositional and First-order logics.
- To trace the inference mechanism in First-order logics.

UNIT - I

Introduction, History, Intelligent Systems, Foundations of AI, Sub areas of AI, Applications. Problem Solving – State-Space Search and Control Strategies: Introduction, General Problem Solving, Characteristics of Problem, Exhaustive Searches, Heuristic Search Techniques, Iterative-Deepening A*, Constraint Satisfaction. Game Playing, Bounded Look-ahead Strategy and use of Evaluation Functions, Alpha-Beta Pruning.

Learning outcomes:

By the end of this unit, the student will be able to

- Define Artificial Intelligence
- How agents work in environments
- Recall uninformed search techniques
- Illustrate the working of informed search techniques

UNIT – II

Logic Concepts and Logic Programming: Introduction, Propositional Calculus, Propositional Logic, Natural Deduction System, Axiomatic System, Semantic Tableau System in Propositional Logic, Resolution Refutation in Propositional Logic, Predicate Logic, Logic Programming. **Knowledge Representation**: Introduction, Approaches to Knowledge Representation, Knowledge Representation using Semantic Network, Extended Semantic Networks for KR, Knowledge Representation using Frames.

Learning outcomes:

By the end of this unit, the student will be able to

- Understand how games improve intellectual abilities of humans
- Choose optimal decisions in games
- Illustrate alpha-beta pruning
- Compare stochastic and partially observable games

UNIT - III

Expert System and Applications: Introduction, Phases in Building Expert Systems, Expert System Architecture, Expert Systems Vs Traditional Systems, Truth Maintenance Systems, Application of Expert Systems, List of Shells and Tools.

Uncertainty Measure – Probability Theory: Introduction, Probability Theory, Bayesian Belief Networks, Certainty Factor Theory, Dempster-Shafer Theory.

Learning outcomes

By the end of this unit, the student will be able to

- Define constraint satisfaction problems.
- Illustrate inference in constraint satisfaction problems
- Contrast backtracking search and local search for constraint satisfaction problems

UNIT - IV

Machine-Learning Paradigms: Introduction, Machine Learning Systems. Supervised, Unsupervised Learning. Inductive Learning. Learning Decision Trees (Text Book 2), Deductive Learning. Clustering, Support Vector Machines. **Artificial Neural Networks**: Introduction, Artificial Neural Networks, Single- Layer Feed-Forward Networks, Multi-Layer Feed-Forward Networks, Radial- Basis Function Networks, Design Issues of Artificial Neural Networks, Recurrent Networks.

Learning outcomes:

By the end of this unit, the student will be able to

- Define knowledge-based agents
- How to represent real-world facts in propositional and first-order logic
- Explain the wumpus world problem-solving process
- Infer proofs in propositional and first-order logic

UNIT - V

Advanced Knowledge Representation Techniques: Case Grammars, **Semantic Web Natural Language Processing:** Introduction, Sentence Analysis Phases, Grammars and Parsers, Types of Parsers, Semantic Analysis, Universal Networking Knowledge.

Learning outcomes:

By the end of this unit, the student will be able to

- Define propositional and first-order inference
- Outline unification and lifting
- Experiment with forward chaining and backward chaining
- Make use of resolution

Text Books:

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

Reference Book:

1. Artificial intelligence by Rich, Knight, Nair, Tata McGraw Hill, 3rd Edition 2009.

Course Outcomes:

- Able to illustrate artificial intelligence, the role of intelligent agents, uninformed and informed search techniques.
- Able to explain competitive environments like game problems.
- Interpret many real-world problems as constraint satisfaction problems.
- Illustrate what knowledge representation is and distinguish propositional and first-order logics.
- Infer proofs using resolution in first-order logic.

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A. -IV SEMESTER

SCA 708 PYTHON PROGRAMMING

Hours per week: 4 End Examination: 60 Marks

Credits 4 Sessionals : 40 Marks

Preamble:

The course is designed to enable the student to write programs for problem solving. After an introduction to R, RStudio, Exploratory Data Analysis, Using R for Data Visualization and Graphics for Communication are designed to work together to make data science fast, fluent. This course lays for developing program logic and for writing programs in Python according to the developed logic.

Course Objectives:

- To explain the elementary programming constructs and file operations and use it in Python programming.
- To describe the concepts like strings, conversion of strings to numbers, lists, tuples, dictionaries and use these in python programming.
- To illustrate the functions, recursive functions and object oriented programming concepts in Python.

UNIT - I

Introduction: Rapid Introduction to Procedural Programming, Creating and Running Python Programs, Data Types, Object References, Collection Data Types, Logical Operations, Control Flow Statements, Arithmetic Operators, Input/Output, Creating and Calling Functions.

Data Types: Identifiers and Keywords, Integral Types, Floating-Point Types, Strings. Collection Data Types: Sequence Types, Set Types, Mapping Types, Iterating and Copying Collections.

Learning outcomes:

By the end of this unit, the student will be able to

- Learn the concepts of programming before actually starting to write programs
- Understand data types, control flow statements
- Know input/output statements and functions
- Develop logic for Problem Solving.

UNIT - II

Control Structures and Functions: Control Structures, Exception Handling, Custom Functions, Lambda Functions. **Modules**: Modules and Packages, Overview of Python's Standard Library, Command-Line Programming. **Object-Oriented Programming:** The Object-Oriented Approach, Custom Classes, Custom Collection Classes.

Learning Outcomes:

By the end of this unit, the student will be able to

- Translate mathematical expressions to R notation using operators
- Construct R programs using various conditional statements
- Develop R programs using loops and nested loops
- Solve problems related to arrays, strings and vectors
- To be made familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc.

- Able to apply the problem solving skills using syntactically simple language

UNIT - III

File Handling: Writing and Reading Binary Data, Writing and Parsing Text Files, Writing and Parsing XML Files, Random Access Binary Files. **Advanced Programming Techniques**: Further Procedural Programming, Further Object-Oriented Programming , Functional-Style Programming. Debugging, Testing and **Profiling**: Debugging, Unit Testing, Profiling.

Learning Outcomes:

By the end of this unit, the student will be able to

- Understand how to read/write to files using python.
- Catch their own errors that happen during execution of programs.
- Learn debugging
- Demonstrate unit testing

UNIT - IV

Processes and Threading: Using the Multiprocessing Module, Using the Threading Module. **Networking**: Creating a TCP Client, Creating a TCP Server. **Regular Expressions**: Python's Regular Expression Language, The Regular Expression.

Learning Outcomes:

By the end of this unit, the student will be able to

- Get an introduction to the concept of Threading.
- Be made familiar with the concepts of Regular Expression
- Connect to creating a TCP Client and creating a TCP Server
- Learn Python's regular expression and language

UNIT - V

Introduction to Parsing: BNF Syntax and Parsing Terminology, Writing Handcrafted Parsers, Pythonic Parsing with PyParsing, Lex/Yacc-Style Parsing with PLY. Introduction to GUI ProgrammingDialog-Style Programs, Main-Window-Style Programs, Creating a Main Window, Creating a Custom Dialog.

Learning Outcomes:

By the end of this unit, the student will be able to

- Get an introduction to the concept of BNF Syntax and Parsing Terminology.
- Made familiar with the concepts of GUI controls and designing GUI applications.
- Able to connect to do the Main-Window-Style Programs.
- Designing and creating a main window and creating a Custom dialog.

Text Book:

1. Programming in Python 3 - A Complete Introduction to the Python Language, Pearson Education, 2nd edition, 2010.

Reference Book:

1. The Fundamentals of Python: First Programs by Kenneth A. Lambert, Cengage Learning, 2011.

Course Outcomes:

- Able to explain the basic program constructs and file operations in Python and express it.
- Design the Python programs using the concepts like strings, conversion of strings to numbers, lists, tuples, dictionaries.
- Implement the functions and object oriented programming concepts in python.
- Create a Graphical User Interface, multiple threads and Client/Server programs in python.
- Able to implement a database connection and CGI programs in python.

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A. - IV SEMESTER
GENERIC ELECTIVE - I
SCA 742 SOFTWARE TESTING

Hours per week: 4 End Examination: 60 Marks

Credits 4 Sessionals : 40 Marks

Preamble:

The software testing is a process of executing a program or application with the intent of finding the bugs. This course will help students learn Catch bugs and break software and test with different testing methods that will help build better software. It will teach and make students think like a software tester and help in finding bugs in code earlier and write better code. The course demonstrates an in-depth understanding of the tools and technologies for software testing and do better programming and test the programs efficiently.

Course Objectives:

- To understand different tools and technologies for software testing.
- To understand different software testing procedure, methods and software testing life cycle.
- To Learn different static and dynamic testing techniques and reviews
- To know how to validate and integrate various regression testing techniques and types.
- To illustrate overall test management and test suite management
- To develop software quality management activaties like quality control, quality assurance and quality management.

UNIT – I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs. **Flow graphs and Path testing**: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

Learning outcomes:

By the end of this unit, the student will be able to

- Define testing and how to evaluate testing.
- Find testing myths and facts of a software environment
- Compare and contrast the various terminologies used in software testing.
- Compare and contrast the verification and validation.
- Apply the testing methodology of a software functionality.

UNIT – II

Transaction Flow Testing: Transaction flows, transaction flow testing techniques.

Dataflow testing: Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

Learning outcomes:

By the end of this unit, the student will be able to

- Demonstrate the black box testing.
- Illustrate the white box testing.
- Contrast static testing and dynamic testing.
- Identify walkthroughs and inspections of a static testing.

- Choose reviews to conduct a testing.

UNIT – III

Domain Testing: Domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

Learning outcomes:

By the end of this unit, the student will be able to

- Apply different types of testing techniques within a software domain
- Choose suitable testing technique to validate the software
- Examine the various types of system testing
- List objectives of regression testing
- Examine the different types of regression testing techniques

UNIT - IV

Paths, Path products and Regular expressions: Path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection. **Logic Based Testing**: Overview, decision tables, path expressions, kv charts, specifications.

Learning outcomes

By the end of this unit, the student will be able to

- Outline the structure of a testing group
- Discover the planning and specification of a testing
- Examine and solve test suite minimization problems
- Choose efficient test suite management
- List test case prioritizations

UNIT - V

State, State Graphs and Transition testing: State graphs, good & bad state graphs, state testing, Testability tips. **Graph Matrices and Application**:Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools.

Learning outcomes:

By the end of this unit, the student will be able to

- How to measure the quality of a product?
- List quality factors
- Choose suitable metrics to provide the quality of a software
- Identify the importance of the model to provide a software quality
- Adapt the process maturity model within a software environment

Text Books:

1. Software Testing techniques by Boris Beizer, Dreamtech Publications, Second edition.
2. Software Testing Tools by K.V.K.K.Prasad, Dreamtech Publications..

Reference Books:

1. The craft of software testing by Brian Marick, Pearson Education.
2. Software Testing by N.Chauhan, Oxford University Press.
3. Introduction to Software Testing by P.Ammann and J.Offutt, Cambridge Univ.Press.

Course Outcomes:

- Acquaint the basic concepts of the software testing
- Able to apply dynamic and static tastings techniques
- Use the different techniques like validation, function, system, acceptance and regression
- Build the structure of testing group ,planning and test suite management
- Familiarize with Software Quality Management concepts.
- Know the concepts of quality cost, quality control and quality assurance

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A.- IV SEMESTER
GENERIC ELECTIVE - I
SCA 744 INTERNET OF THINGS

Hours per week: 4 End Examination: 60 Marks

Credits 4 Sessionals : 40 Marks

Preamble:

The digital space is major transformations in the last couple of years and as per industry experts would continue to evolve itself. The latest entrant to the digital space is the Internet of Things (IoT). IoT can also be defined as interplay for software, telecom and electronic hardware industry and promises to offer great opportunities for much industry. This course will make the student learn as with the advent of the Internet of Things (IoT), fed by sensors soon to number in the trillions, working with intelligent systems in the billions, and involving millions of applications.

Course Objectives:

- To assess the vision and introduction of IoT.
- Understand the division of network functionalities into layers.
- Be familiar with the components required and Understand IoT Market perspective.
- To Implement Data and Knowledge Management and use of Devices in IoT technology.
- To build different types of networks and learn the functionalities of each layers.

UNIT - I

What is the Internet of Things: Overview and Motivations, Examples of Applications,IPv6 Role, Areas of Development and Standardization, Scope of the Present Investigation.

Internet of Things Definitions And Frameworks: IoT Definitions IoT frameworks, Basic Nodal capabilities.

Learning outcomes:

By the end of this unit, the student will be able to

- Understand the concepts of Internet of Things
- Understanding the function blocks of IOT
- Observing the difference between IOT and M2M

UNIT - II

Internet of Things Applications Examples: e-Health/Body area Networks, City automation, Home Automations, Smart cards.

Fundamental IoT Mechanisms: Identification of IoT Objectives and Services, Structural Aspects of the IoT, Key IoT Technologies.

Evolving IoT Standards: Overview and Approaches, IETF IPv6 Routing Protocol for RPL Roll, Constrained Application Protocol (CoAP), Representational State Transfer(REST), ZigBee IP(ZIP), IP in Smart Objects.

Learning outcomes:

By the end of this unit, the student will be able to

- Analyze basic protocols in wireless sensor network
- Understand the process of sensors
- Understanding how a node in discovered after deployment
- Analyzing techniques for data aggregation and dissemination

UNIT - III

Layer ½ Connectivity: Wireless Technologies for the IoT: WPAN Technologies for IoT/M2M, Cellular and Mobile Network Technologies for IoT/M2M.

Learning outcomes:

By the end of this unit, the student will be able to

- Understanding the management of IOT devices
- Observing the Technologies for M2M.
- Analysing Cellular and mobile network technologies in IOT
- Understanding the security challenges in IOT

UNIT - IV

Layer 3 Connectivity, IPv6 Technologies for the IoT: Overview and Motivations, Address Capabilities, IPv6 Protocol Overview,IPv6 Tunneling, IPSec in IPv6, Header Compression Schemes, Quality of Service in IPv6, Migration Strategies to IPv6.

Learning Outcomes:

By the end of this unit, the student will be able to

- Implement basic IOT applications on embedded platform
- To develop and deploy basic and complex IOT networks
- Learning IOT concepts with migration strategies

UNIT - V

Layer 3 Connectivity, Mobile IPv6 Technologies for the IoT : Overview, Protocol Details, Generic Mechanisms, New IPv6 Protocol, Message Types and Destination Option, Modifications to Ipv6 Neighbor Discovery, Requirements for Various IPv6 Nodes, Correspondent Node Operation, Mobile Node Operation, Relationship to IPv4 Mobile IPv4(MIP)

Learning Outcomes:

By the end of this unit, the student will be able to

- Design IOT applications in different domain and be able to analyze their performance
- Understanding the role of IOT in home automation
- IOT industry application review

Text Books:

1. Daniel Minoli, Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications, 2015, Wiley Publications.

Reference Books:

1. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things: Key Applications and Protocols, 2nd Edition, Willy Publications.
2. Daniel Kellmereit, Daniel Obodovski, "The Silent Intelligence: The Internet of Things",. Publisher: Lightning Source Inc; 1st edition, 2014

Course Outcomes:

- Able to understand the application areas of IOT from a global context.
- Determine the Market perspective of IoT.
- Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
- Able to understand building blocks of Internet of Things and characteristics.
- Illustrate the application of IoT in Industrial Automation and identify Real World Design Constraints.

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A.- IV SEMESTER
GENERIC ELECTIVE - I
SCA 746 ENTERPRISE RESOURCE PLANNING

Hours per week: 4 End Examination: 60 Marks

Credits 4 Sessionals : 40 Marks

Preamble:

Enterprise Resource Planning is the latest high end solution; information technology has lent to business application. The ERP solutions seek to streamline and integrate operation processes and information flows in the company to synergize the resources of an organization namely men, material, money and machine through information. This course will give insight on ERP issues, risks and solutions.

Course Objectives:

- To understand the steps and activities in the ERP life cycle
- To describe the typical functionality in an ERP system,
- To provide fundamental knowledge regarding the concepts
- To design the structure of ERP systems
- To understand the skill requirement for implementation of ERP in a business enterprise.

UNIT - I

ERP Concepts: Enterprise System, Evolution of ERP, tangible and intangible benefits, Emerging trends in adoption of ERP, Key issues and risks in ERP.

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn the basic concepts of ERP
- Understand the basic concepts of ERP
- Analyze the benefits of targetable and intangible
- Understand the trends and key issues in ERP

UNIT - II

Business Modules in an ERP Package: Finance, Manufacturing, Human resources, Plant maintenance, Materials management, Quality management, Sales and Distribution, ERP market, SAP AG, People soft, Baan, JD Edwards, Oracle, QAD, SSA.

Learning Outcomes:

By the end of this unit, the student will be able to

- Design the modules in ERP Package.
- learn and identify the different technologies used in ERP.
- To learn the concepts of ERP Manufacturing Perspective and ERP Modules.
- To learn what are the benefits of ERP

UNIT - III

Need Analysis: Competitive environment analysis, Gap analysis, Cost elements, Feasibility analysis, ERP industries verticals, ERP architecture, ERP software, ERP package evaluation criteria, Package life cycle, Functional requirement specification, Request for proposal, Vendor selection, ERP consultants.

Learning Outcomes:

By the end of this unit, the student will be able to

- Understand and apply the concepts of ERP Manufacturing Perspective and ERP Modules.
- Discuss the benefits of ERP
- To analyze the Competitive environment analysis that should be related to industry.
- Apply different tools used in ERP.

UNIT - IV

ERP Implementation Lifecycle: Implementation methodology, implementation strategies - transition, big bang, phased, parallel, process line, hybrid- implementation approaches.

Learning Outcomes

By the end of this unit, the student will be able to

- Understand and implement the ERP life cycle.
- Implement the different strategies and methodologies in ERP Life cycle.
- Identify different approaches used in ERP.

UNIT-V

Organizational Transformational Model of ERP Success: Cross functional, Organizational and Industrial impacts, Capability maturity framework, Future directions and trends in ERP. Case Analysis

Learning Outcomes:

By the end of this unit, the student will be able to

- learn the different tools used in ERP.
- Understand the capability maturity frame work
- Implement the organizational Transformational Model of ERP
- Analyze the industrial impact in different frame work

Text Book:

1. EnterpriseResource Planning Demystified by AlexisLeon, TataMcGrawHill Publishing Company Ltd., 2014.

Reference Book:

1. A Textbook of Enterprise Resource Planning by Mahadeo Jaiswal and Ganesh Vana, First edition.

Course Outcomes:

- Able to analyze various ERP in the real world and apply the various methodologies for the ERP.
- Able to describe the role of Business Modules in an ERP Package
- Understand the Competitive environment analysis.
- Design and Implementation methodology strategies in different approaches.
- Apply Information Technology and ERP for future directions.

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A.- IV SEMESTER
SAE 722 TECHNICAL COMMUNICATION SKILLS LAB

Hours per week: 4 Examination: 100 Marks

Credits: 2

Course Objectives:

The language Lab focuses computer-aided multi-media instruction and language acquisition to achieve the following targets:

- To expose the students to a variety of self-instructional, learner-friendly modes of languages learning.
- To enable them to learn better pronunciation through stress on word accent, intonation, and rhythm.
- To train the students to use language effectively to face interviews, group discussions, public speaking.
- To initiate them into greater use of the computer in resume preparation, report writing, format-making etc.

1. Introduction to Phonetics
2. Introduction to Vowels and Consonants and associated Phonetic symbols
3. Introduction to Accent
4. Introduction to Intonation and Rhythm
5. Listening for General Content
6. Listening Specific Content
7. Situational Dialogues / Role Play
8. Just a Minute sessions
9. Public Speaking
10. G.D/Debate
11. Mock Interviews

Reference Books:

1. Developing Communication Skills by Krishna Mohan & Meera Benerji, Macmillan.
2. Speaking English Effectively by Krishna Mohan & NP Singh, Macmillan.
3. Better English Pronunciation by JDO Connor, UBS-Cambridge.
4, Oxford Practice Grammar with Answers, John Eastwood, Oxford.
5.Handbook of English Grammar and Usage by , Mark Leaster and Larry Beason, TMH.
6. A text book of English Phonetics for Indian Students by T. Blalsubramanian, Macmillan.
7. Lingua TOEFL CBT Insider by Dreamtech.
8.TOEFL & GRE(KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
9.English Skills for Technical Students, WBSCTE with British Council
10.A Handbook of English for Competitive Examinations by B Shyamala Rao, Blakie Books.

Course Outcomes:

By the end of this course, the student will be able to:

- Familiarize with phonetic symbols
- Utilize phonetic dictionary symbols to continue to improve pronunciation
- Apply the rules for word stress
- Identify and properly place stress on the right syllable of the word
- Produce native-like intonation, rhythm and stress in sentences
- Take organized notes on lectures and listening passages
- Identify speaker's purpose and tone
- Make inferences about spoken discourse- listening to audio
- Don the role assigned effectively with spontaneous and sparkly dialogues
- Speak fluently without any pauses, hesitation or repetition
- Address public without fear and diffidence
- Acquire the skills needed for a G.D and participate efficiently
- Face interviews with confidence and required fluency in English

Prepared by: Dr. Sushma Raj Verified by : Prof. V. NagaLakshmi

M.C.A. – IV SEMESTER

SCA 722 DATA MINING LAB USING R

Hours per week: 4 Examination: 100 Marks

Credits: 4

1. Introductory commands in R
2. Programs using Descriptive Statistics.
3. Correlation and Regression Analysis.
4. Program to demonstrate pre-processing on dataset Mtcars
5. Program to demonstrate Association rules on Groceries dataset using Apriori Algorithm
6. Program to demonstrate Classification Rules process on dataset Titanic using id3 Algorithm .
7. Program to demonstrate Classification rule process on dataset Titanic using CART Algorithm.
8. Program to demonstrate Classification rule process on attrition dataset using Naïve Bayesian Algorithm.
9. Program to demonstrate Clustering rule process on dataset Iris using simple k-means .
10. Program to demonstrate Clustering rule process on dataset Iris using Hierarchical Clustering.
11. Program to demonstrate Outlier Detection using dataset Iris.
12. Construction of the fact and Dimension Tables for the Multi dimensional data.
13. Perform different measures on the data warehouse data.

Reference Books:

1. Data Mining Concepts and Techniques by Jiawei Han Micheline Kamber, and Jian Pei, Elsevier Publications, 3rd Edition, 2013.
2. Statistics Using R by Sudha G.Purohit, Narosa Publications, 2015.

Course Outcomes:

- The student will learn how to solve correlation and regression problems using R.
- The student with existing datasets will do preprocessing of the data.
- The student will be able to apply various classification algorithms on the data sets by using R commands.
- The student will acquire knowledge about various clustering algorithms and its use.
- The student will come to know about fact and dimensional table and various operations on warehouse data.

Prepared by: Dr. M. Sesha Shayee Verified by : Prof. V. NagaLakshmi

M.C.A. – IV SEMESTER
SCA 724 PYTHON PROGRAMMING LAB

Hours per week: 4 Examination: 100 Marks
Credits: 2

1. Strings:

a) Write a Python program to calculate the length of a string

b) Write a Python program to count the number of characters (character frequency) in a string. Sample String : google.com' Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}

c) Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.
Sample String: 'w3resource' Expected Result: 'w3ce'
Sample String: 'w3' Expected Result: 'w3w3'
Sample String: ' w' Expected Result: Empty String

d) Write a Python script that takes input from the user and displays that input back in upper and lower cases.

e) Write a Python program to count the occurrences of each word in a given sentence.

f) Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).

2. Lists:

a) Write a Python program to sum all the items in a list

b) Write a Python program to get the largest number from a list.

c) Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.
Sample List : ['abc', 'xyz', 'aba', '1221'] Expected Result : 2

d) Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.
Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)] Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

e) Write a Python program to count the number of elements in a list within a specified range

f) Write a Python program using Sieve of Eratosthenes method for computing primes upto a specified number

g) Write a Python program to convert list to list of dictionaries.
Sample lists: ["Black", "Red", "Maroon", "Yellow"], ["#000000", "#FF0000", "#800000", "#FFFF00"]
Expected Output: [{'color_name': 'Black', 'color_code': '#000000'}, {'color_name': 'Red', 'color_code': '#FF0000'}, {'color_name': 'Maroon', 'color_code': '#800000'}, {'color_name': 'Yellow', 'color_code': '#FFFF00'}]

h) Write a Python program to split a list every Nth element.

i) Sample list: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n']
Expected Output: [['a', 'd', 'g', 'j', 'm'], ['b', 'e', 'h', 'k', 'n'], ['c', 'f', 'i', 'l']]

j) Write a Python program to find the list in a list of lists whose sum of elements is the highest. Sample lists: [1,2,3], [4,5,6], [10,11,12], [7,8,9] Expected Output: [10, 11, 12]

k) Write a Python program to remove duplicates from a list of lists.
Sample list : [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]
New List : [[10, 20], [30, 56, 25], [33], [40]]

3. Dictionary:

a) Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys
Sample Dictionary -
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}

b) Write a Python program to combine two dictionary adding values for common keys.
d1 = {'a': 100, 'b': 200, 'c':300}
d2 = {'a': 300, 'b': 200, 'd':400}
Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

c) Write a Python program to count the values associated with key in a dictionary.
Sample data: = [{'id': 1, 'success': True, 'name': 'Lary'}, {'id': 2, 'success': False, 'name': 'Rabi'}, {'id': 3, 'success': True, 'name': 'Alex'}]
Expected result: Count of how many dictionaries have success as True

d) Write a Python program to match key values in two dictionaries.
Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2}
Expected output: key1: 1 is present in both x and y

4. Files:

a) Write a Python program to read first n lines of a file.
b) Write a Python program to read a file line by line and store it into a list.
c) Write a Python program to combine each line from first file with the corresponding line in second file.
d) Write a Python program to assess if a file is closed or not.
e) Write a Python program to remove leading zeros from an IP address

5. BST:

a) Write a Python program to create a Balanced Binary Search Tree (BST) using an array (given) elements where array elements are sorted in ascending order.
b) Write a Python program to delete a node with the given key in a given Binary search tree (BST).
Note: Search for a node to remove. If the node is found, delete the node.
c) Write a Python program to convert a given array elements to a height balanced Binary Search Tree (BST).
d) Write a Python program to sort a list of elements using Selection sort.

6. Classes:

a) Write a Python class to find the three elements that sum to zero from a set of n real numbers. -
Input array : [-25, -10, -7, -3, 2, 4, 8, 10]
Output : [[-10, 2, 8], [-7, -3, 10]]

b) Write a Python class to reverse a string word by word.

7. **Maths:**
   a) Write a Python program to compute Euclidean distance in Python
   b) Write a Python program to implement Euclidean Algorithm to compute the greatest common divisor (gcd)
   c) Write a Python program to create a range for floating numbers.

Reference Books:

1. Programming in Python 3 - A Complete Introduction to the Python Language, Pearson Education, 2nd edition, 2010.
2. The Fundamentals of Python: First Programs by Kenneth A. Lambert, Cengage Learning, 2011.

Course Outcomes:

- Able to learn introductory concepts of Python programming
- Practice strings and lists
- Demonstrate on dictionary concepts
- Able to work with data structures and files using Python language
- Design programs to work with classes
- Develop programs to perform mathematical calculations

Prepared by: Ms. K. Vanitha Verified by : Prof. V. NagaLakshmi

M.C.A.-V SEMESTER
SCA 801 CLOUD COMPUTING

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Cloud computing is a model for enabling ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources. Cloud computing enables increasing number of IT services to be delivered over the Internet. The cloud platform enables business to run successfully without dedicated hardware, software and services. This course will make the student learn the concepts of cloud computing and will understand how to run applications on cloud.

Course Objectives:

- To learn the concept of cloud computing
- To understand the concepts of virtualization
- To know the different services provided by cloud
- To analyze the cloud computing architecture
- To demonstrate on different types of clouds i.e., AWS and Google

UNIT – I

Introduction: Cloud Computing at a Glance, The Vision of Cloud Computing, defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments, Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments, Application Development, Infrastructure and System Development, Computing Platforms a Technologies, Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka. (8)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn the introductory concepts of cloud
- Analyze the characteristics and benefits of cloud
- Differentiate service oriented computing and utility oriented computing
- Understand cloud computing environment
- Know the different cloud providers i.e., AWS, Google AppEngine, MS Azure etc.,

UNIT - II

Virtualization: Introduction, Characteristics of Virtualized Environments, Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples, Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V. (8)

Learning Outcomes:

By the end of this unit, the student will be able to

- Know the characteristics of virtualized environment
- Understand virtualization techniques
- Analyze the Pros and Cons of virtualization
- Learn different types of virtualization techniques
- Illustrate VMware

UNIT - III

Cloud Computing Architecture: Introduction Cloud Reference Model, Architecture Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Dentition, Cloud Interoperability and Standards, Scalability and Fault Tolerance, Security, Trust, and Privacy, Organizational Aspects. (8)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn cloud reference model
- Understand Architecture
- Analyze different services
- List different types of clouds
- Know the security and privacy issues

UNIT - IV

Discovering the AWS Development Environment : Starting Your AWS Adventure, Defining the AWS Cloud, Discovering IaaS, Determining Why You Should Use AWS, Considering the AWS-Supported Platforms.

Obtaining Development Access to Amazon Web Services: Discovering the Limits of Free Services, Considering the Hardware Requirements, Getting Signed Up, Testing Your Setup, Choosing the Right Services, Getting a Quick Overview of Free-Tier Services, Matching AWS Services to Your Application, Considering AWS Security Issues. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn introduction to AWS cloud
- Understand how it provides IaaS\
- Develop its services
- Know the supported platforms
- Analyze Security issues

UNIT - V

Starting the Development Process: Considering AWS Communication Strategies, Defining the Major Communication Standards, Understanding How REST Works, Creating a Development Environment, Choosing a Platform, Obtaining and Installing Python, Working with the Identity and Access Management Console, Installing the Command Line Interface Software, Configuring S3 Using CLI, Configuring S3 Using Node.js, Configuring S3 Using a Desktop Application, Creating a Virtual Server Using EC2, Getting to Know the Elastic Compute Cloud (EC2), Working with Elastic Block Store (EBS) Volumes, Discovering Images and Instances.

Performing Basic Development Tasks : Understanding AWS Input/Output, Considering the Input /Output Options, Working with JSON, Working with XML, Working with Amazon API Gateway, Developing Web Apps Using Elastic Beanstalk, Considering Elastic Beanstalk (EB) Features, Deploying an EB Application, Updating an EB Application, Removing Unneeded Applications, Monitoring Your Application Using Amazon CloudWatch. (12)

Learning Outcomes:

By the end of this unit, the student will be able to

- Understands how REST works
- Develops environment by choosing the platform
- Configures S3 using CLI, using Node.js and using Desktop application
- Know Elastic Cloud Computing
- Practices working with JSON, XML and Amazon API

Text Books:

1. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013.
2. AWS for Developers- Dummies by John Paul Mueller, John Wiley & Sons Inc. publications, 2017.

Reference Books:

1. Cloud Computing Concepts Technology Architecture by Thomas Erl, Pearson Education, 2014.
2. Cloud Computing Explained by John Rhoton , Recursive Press, 2009.

Course Outcomes:

- Able to know the introductory concepts of cloud computing
- Understands the services provided by the cloud
- Learn Service oriented computing and Utility computing
- Able to demonstrate different virtualization techniques
- Know how to obtain access to AWS
- Demonstrates how to work with JSON, XML and Amazon API

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.-V SEMESTER
SCA803 CYBER SECURITY

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Cyber space is a complex environment consisting of interactions between people, software and services, supported by worldwide distribution of information and communication technology (ICT) devices and networks. This course deals with concepts, measures and algorithms for the protection of information infrastructure and preservation of the confidentiality, integrity and availability of information against cyber threats and different types of malware

Course Objectives:

- To make students understand different types of attacks and malware
- To make the student learn symmetric algorithms, asymmetric algorithms and hash functions as measures against attacks
- To know the concepts of key distribution
- To illustrate different mechanisms implemented for e-mail, web and IP security
- To model PGP and SSL
- To relate firewalls, cyber threats and their defense

UNIT – I

Cyber Security Overview: Introduction, Security from a Global Perspective, Trends in the Types of Attacks and Malware, the types of Malware, Vulnerability Naming schemes and Security configuration settings, Common Vulnerabilities and Exposures(CVE), Common Configuration Enumeration (CCE), Zero-Day vulnerabilities, Network and Information Infrastructure Defense Overview.

Symmetric Encryption and Message Confidentiality: Symmetric Encryption Principles, Symmetric Block Encryption Algorithms, DES, AES, Stream Ciphers and RC4. (12)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn introductory concepts of Cyber Security
- Understand different types of attacks and malware
- Know different vulnerabilities
- Find symmetric encryption principles
- Analyze different symmetric algorithms

UNIT-II

Public-Key Cryptography and Message Authentication: Approaches to Message Authentication, Secure Hash Functions, Message Authentication Codes, Public Key Cryptography Principle, Public-Key Cryptography Algorithms, RSA, Diffie Hellman key Exchange, Digital Signatures.

Key Distribution and User Authentication: Symmetric Key Distribution Using Symmetric Encryption, Kerberos, Key Distribution Using Asymmetric Encryption, X.509 Certificates, Public Key Infrastructure. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- Learn concepts of message authentication and different types of authentication methods
- Find Public key cryptography principles
- Analyze different asymmetric algorithms
- Illustrate key distribution using Kerberos and asymmetric encryption
- Summarize public key infrastructure

UNIT - III

Transport-Level Security: Web Security Issues, Secure Sockets Layer (SSL), Transport Layer Security (TLS)

Electronic Mail Security: Pretty Good Privacy (PGP), S/MIME, **IP Security:** IP Security Overview, IP Security Policy, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- List web security issues
- Illustrate different mechanisms implemented for e-mail i.e., PGP and S/MIME
- Explain IP Security Policy
- Analyze Encapsulating Security Payload
- Summarize Internet Key Exchange concepts

UNIT – IV

Firewalls: Overview, Unified Threat Management, Firewalls, Stateless Packet Filtering, Stateful/Session Filtering, Application-Level Gateways, Circuit-Level Gateways, A comparison of Four Types of Firewalls, The Architecture for a Primary-Backup Firewall, The Cisco Firewall as an Enterprise Firewall, The Small office/Home office Firewall, Emerging Firewall Technology. (8)

Learning Outcomes:

By the end of this unit, the student will be able to

- Outline threat management
- Learn firewalls, stateless and stateful filtering
- Compare different types of firewalls
- Explain primary architecture of firewall
- Classify firewalls as enterprise, small office/home

UNIT – V

Intrusion Detection/Prevention System: Overview, IDS/IPS Building Blocks, Host-Based or Network-Based IDS/IPS, Host-Based or Network-Based IDS/IPS, The Approaches Used for IDS/IPS, Honeypots, The Detection of Polymorphism/Metamorphic Worms, Distributed Intrusion Detection Systems and Standards.

Cyber Threats and Their Defense: Domain Name System (DNS) Protection, Router Security, Spam/Email Defensive Measures, Phishing Defensive Measures, Web-Based Attacks, Database Defensive Measures, Botnet Attacks and Applicable Defensive Techniques. (10)

Learning Outcomes:

By the end of this unit, the student will be able to

- Know introductory concepts of Intrusion detection/prevention
- Differentiate Host-Based or Network-Based IDS/IPS
- Analyze the approaches used for IDS/IPS
- Learn Domain Name System Protection
- Illustrate Email and Phishing Defensive Measures

Text Books:

1. Introduction to Computer Networks and Cyber Security, Chwan-Hwa (John) Wu, J.David Irwin, CRC Press, Taylor & Francis Groups, 2014 (UNIT – I, IV and V)
2. Network Security Essentials, Applications and Standards, 5th edition, William Stallings, Pearson Education, 2013. (UNIT– II and III)

Reference Books:

1. Cryptography & Network Security, Behrouz A. Forouzan, Tata McGraw-Hill, New Delhi, 2007.
2. Network Security: Private Communication in a Public World, Kaufman, Pearson Education Asia, New Delhi, 2002.
3. Cryptography and Network Security: Atul Kahate, Tata McGraw-Hill Publishing Company Limited, New Delhi, 2003.

Course Outcomes:

- Identify different types of threats and malware
- Analyze different symmetric key encryption algorithms
- Learn and Identify different applications of symmetric and asymmetric key algorithms
- Summarize different security mechanisms
- Explain threat management and firewalls
- Illustrate different intrusion detection and prevention mechanisms

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.-V SEMESTER
SCA 805 MACHINE LEARNING

Hours per week: 4 End Examination:60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Machine learning is about designing programs that can learn without being explicitly programmed. It is a branch of Artificial Intelligence in which the student will learn concepts/patterns/hypotheses from Data by using heuristic based algorithms. This field has become so popular that one can find machine leaning applications in virtually all domains ranging from identifying emails as spam or legitimate to automated vehicle guided system to game playing to credit card fraud detection

Course Objectives:

- The student will get to know the basic concepts from statistics, artificial intelligence, information theory, and other disciplines as the need arises.
- To learn machine learning techniques and those concepts most relevant to machine learning.
- To understand the decision tree learning algorithms.
- To understand hypotheses evaluation and Bayesian learning theorem.
- To understand computational learning theorem.

UNIT - I

Introduction to Machine Learning : Well-Posed Learning Problem, Designing A Learning System, Perspectives And Issues In Machine Learning.

Concept Learning And The General-To-Specific Ordering: Introduction, A Concept Learning Task, Concept Learning as Search, Find-S: Finding A Maximally Specific Hypothesis, Version Spaces And The Candidate Elimination Algorithm, Remarks On Version Spaces and Candidate - Elimination, Inductive Bias. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn well-posed learning problems
- Know the issues involved in machine learning
- Apply maximally Specific Hypothesis
- Analyze version spaces and candidate elimination algorithm

UNIT - II

Decision Tree Learning: Introduction, Decision Tree Representation, Appropriate Problems For Decision Tree Learning, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Learning, Inductive Bias in Decision Tree Learning, Issues in Decision Tree Learning. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define Decision Tree

- Demonstrating decision tree representation
- Apply the basic decision tree algorithm
- Analyze the inductive bias in decision tree algorithm
- To find out the issues in decision tree algorithm

UNIT - III

Evaluating Hypotheses: Motivation, Estimating Hypothesis Accuracy, Basics of Sampling Theory , A General Approach For Deriving Confidence Intervals, Difference In Error of Two Hypotheses, Comparing Learning Algorithms. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define the basic of sampling theory
- Demonstrate the basis of sampling theory
- Apply the basic decision tree algorithm
- Compare the differences in error of two hypotheses
- Compare learning algorithms

UNIT - IV

Bayesian Learning: Introduction, Bayes Theorem, Bayes Theorem and Concept Learning, Maximum Likelihood And Least-Squared Error Hypothesis, Maximum Likelihood Hypothesis For Predicting Probabilities, Minimum Description Length Principle, Bayes Optimal Classifier, Gibbs Algorithm, Naive Bayes Classifier, An Example: Learning to Classify Text, Bayesian Belief Networks, The EM Algorithm. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Definition of Bayes Theorem
- Explain maximum likelihood and least square error
- Apply Gibbs algorithm and Naïve Bayes Classifier
- Compare the differences in error of two hypotheses
- Choose the best algorithm for the application

UNIT-V

Computational Learning Theory: Introduction, Probably Learning An Approximately Correct Hypothesis, Sample Complexity for Infinite Hypothesis Spaces, The Mistake Bound Model Of Learning. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Illustrate an appropriate hypothesis
- Identify infinite hypothesis spaces
- Analyze the mistake bound mode of learning

Text Book:

1. Machine Learning by Tom Mitchell, Mc Grawhill Publications, 2013.

Reference Books:

1. Machine Learning A Probabilistic Perspective by Kevin P.Murthy, MIT Press 2012.
2. Introduction to Machine Learning by Ethen Alpaydin, MIT Publishers, 2nd Edition, 2010.

Course outcomes:

- Able to learn various machine learning algorithms
- Analyze version spaces and candidate elimination algorithm
- Able to evaluate hypotheses
- Able to practice Baysian learning
- Design an algorithm to classify text
- Able to explain computational learning theory

Prepared by: Dr. M. Sesha Shayee Verified by : Prof. V. NagaLakshmi

M.C.A.- V SEMESTER
GENERIC ELECTIVE - II
SCA 841 DevOps

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals: 40 Marks

Preamble:

DevOps is a cross-disciplinary community of practice dedicated to the study of building, evolving and operating rapidly-changing resilient systems at scale and it is the practice of operations and development engineers participating together in the entire service life cycle, from design through the development process to production supports. This course will help the students how to develop applications by integrating different tools using cloud services.

Course Objectives:

- To learn how to develop applications by integrating different tools.
- To understand Kubernetes which is a portable, extensible open-source platform for managing containerized workloads and services
- To Illustrate cluster administration
- To experiment kubernetes on AWS
- To experiment kubernetes on Google cloud

UNIT - I

Introduction to Devops: Software delivery challenges, Waterfall and physical delivery, Agile and electrical delivery, software delivery on the cloud, continuous Integration, continuous Integration, Continuous Delivery, Configuration management, Infrastructure as code, Orchestration, Trend of Microservices, Modular programming, package management, MVC design pattern, Monolithic application, Remote Procedure call, RESTful design, Microservices. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Know software delivery challenges and different delivery models
- Understand delivery on cloud and continuous integration
- Illustrate microservices and package management
- Explain MVC design pattern and Monolithic application
- Apply Remote Procedure and RESTful design

UNIT - II

Automation and Tools: Continuous integration tool, Continuous Delivery tool, Monitoring and logging tool, Communication tool, Public cloud. DevOps with Container, understanding container, Resource isolation, Linux container concept, Containerized delivery, getting started container, Installing Docker for Ubuntu, Installing Docker for CentOS, Installing Docker for macOS. Container life cycle, Docker basics, Layer, image, container, and volume, distributing images, connect container, working with Dockerfile, writing your first Dockerfile, Dockerfile syntax, Organizing a Dockerfile, Multi-containers orchestration, Piling up containers, Docker compose overview, Composing containers. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn integration, continuous delivery, Monitoring &logging and communication tools
- Know Public cloud with Devops
- Understand container, resource isolation
- Installing Docker for Linux, Ubuntu, CentOS and MacOS
- Experimenting Docker file syntax, organizing Docker file

UNIT - III

Getting started with Kubernetes: Understanding Kubernetes, Kubernetes components, Master components, API server, Controller, etc, Scheduler, Node components, Kubelet, Proxy, Docker, Interaction between kubernets master and nodes. Preparing the environment, kubectl, kubernetes resources, kubernets objects, Namespace, Name, Label and selector, Annotation, Pods, ReplicaSet(RS) and Replication Controller (RC), Deployments, Services, volumes, Secrets, Control Map, Using ConfigMap via volume, Using ConfigMap via environment variables, Multi-containers orchestration. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Understand Kubernetes, its components, API server and Scheduler
- Learn Proxy and the interaction between kubernetes master and nodes.
- Develop environment kubectl, kuberetes resources and objects.
- Experiment on kubernetes objects, Namespace, Label and selector.
- Design Control Map and Config Map

UNIT - IV

Monitoring and Logging: Inspecting a container, Kubernetes dashboard, Montoring in Kubernetes, Application, Host, External resources, container, Kubernetes, Getting monitoring essentials for Kubernetes, Hands-on monitoring, Meeting Prometheus

Cluster Administration: Kubernetes namespaces, Default namespaces, Create a new namespace, Create a Job, Create POD, Monitor POD, Monitor Cluster Hosts, Monitoring Host Logs, Understand the Manual Scaling, Self Healilng. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn how to inspect a container, dash board.
- Understand how to monitor kubernetes applications, host and external resources
- Analyze cluster administration
- Illustrate how to create a job, a pod
- Analyze how to monitor PODs, Hosts and Logs

UNIT - V

Kubernets on AWS: Introduction to AWS, Public cloud, API and infrastructure as code, AWS components, Setup kubertenetes on AWS, Install kops, Run kops, Kubernetes cloud provider, Maintenance Kubernetes cluster by kops.

Kubernetes on GCP (Google Cloud) : Create K8s Cluster on GCP, Deployinginix on K8s Cluster, Verify the App via Public IP, Understand GUI & Cloud Shell. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Learn introductory concepts of AWS
- Understand Public cloud, API and AWS components
- Experiment kubernetes on AWS
- Install and Run kops
- Deploy K8s on Google Cloud

Text Book:

1. DevOps with Kubernetes: Accelerating software delivery with container by Hideto Saito, Hui-Chuan Chloe Lee, Cheng-Yang Wu, O' Reilly publications, 2017.

Reference Book:

1.Managing Kubernetes:Operating Kubernetes Clusters in the Real Worls by Brendan Burns, Craig Tracey, O'Reilly publications, 2017.

Course Outcomes:

- Able to understand Kubernetes which is a portable, extensible open-source platform for managing containerized workloads and services
- Able to explain cluster administration
- Able to handle Docker files and its organization
- Able to create a job and a pod
- Able to experiment kubernetes on AWS and google

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.- V SEMESTER
GENERIC ELECTIVE - II
SCA 843 BIG DATA ANALYTICS

Hours per week: 4 End Examination: 60 Marks

Credits: 4 Sessionals: 40 Marks

Preamble:

Big data analytics is the often complex process of examining large and varied data sets or big data to uncover information including hidden patterns, unknown correlations, market trends and customer preferences that can help organizations make informed business decisions.

Course Objectives:

- To understand what big data is.
- To learn about Data Analytics Life cycle.
- To do Data planning, data preparation and performing ETLT.
- To learn about HDFS, Map reduce and Yarn.
- To give a overview to the student regarding Hbase, Hive and Mahout.
- To develop various frameworks and Appliances for Graph Analytics

UNIT - I

Introduction to Big Data Analytics: Big Data Overview, Data Structures, Analyst Perspective on Data Repositories, State of the Practice in Analytics, Bl Versus Data Science, Current Analytical Architecture, Drivers of Big Data, Emerging Big Data Ecosystem and a New Approach to Analytics , Key Roles for the New Big Data Ecosystem, Examples of Big Data Analytics.

Data Analytics Lifecycle -Data Analytics Lifecycle Overview, Key Roles for a Successful Analytics Project , Background and Overview of Data Analytics Lifecycle, Discovery , Learning the Business Domain, Resources, Framing the Problem, identifying Key Stakeholders, Developing Initial Hypotheses ,Identifying Potential Data Sources, Data Preparation ,Preparing the Analytic Sandbox , Performing ETLT , Learning About the Data, Data Conditioning, Survey and Visualize, Common Tools for the Data Preparation, Planning , Data Exploration and Variable Selection, Model Selection, Common Tools for the Model Planning Phase , Model Building, Common Tools for the Model/Building Phase, Communicate Results, Operationalize.

(12)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define Big Data
- Illustrate various Data Repositories
- Identify Potential Sources
- Analyze common Tools for Data Preparation
- Discuss common tools for model /building phase

UNIT - II

Review of Basic Data Analytic Methods Using R: Introduction to R, R Graphical User Interfaces, Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis, Visualization before Analysis, Dirty Data, Visualizing a Single Variable, Examining Multiple Variables, Data Exploration versus Presentation, Statistical Methods for Evaluation, Hypothesis Testing, Difference of Means, Wilcoxon Rank-Sum Test, Type I and Type II Errors, Power and Sample Size (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Illustrate Graphical User Interface and concepts in R.
- Explore Data Analysis
- Statistical methods for Evaluation
- Hypothesis testing
- Explain Difference of Mean

UNIT - III

Big Data Tools and Techniques: Understanding Big Data Storage, A General Overview of High-Performance Architecture, HDFS, Map Reduce and YARN, Expanding the Big Data Application Ecosystem, Zookeeper, HBase, Hive, Pig, Mahout. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Understand Big Data Storage
- Overview of the Architecture
- Illustrate HDFS, Map Reduce and Yarn
- Applying Big data Ecosystem

UNIT - IV

Developing Big Data Applications, Parallelism, The Myth of Simple Scalability, The Application Development Framework, The Map Reduce Programming Model, A Simple Example, More on Map Reduce, Other Big Data Development Frameworks, The Execution Model. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Spell the myth if simple scalability
- Illustrate the application development framework
- Simple example on map reduce
- Build an evaluation model

UNIT - V

NoSQL Data Management for Big Data-What is NoSQL?, Schema-less Models, Increasing Flexibility for Data Manipulation, Key—Value Stores, Document Stores, Tabular Stores, Object Data Stores, Graph Databases.

Using Graph Analytics for Big Data-What Is Graph Analytics?, The Simplicity of the Graph Model, Representation as Triples, Graphs and Network Organization, Choosing Graph

Analytics, Graph Analytics Use Cases, Graph Analytics Algorithms and Solution Approaches, Technical Complexity of Analyzing Graphs, Features of a Graph Analytics Platform, Dedicated Appliances for Graph Analytics (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Listing the graph analytics
- Illustrating graphs and network organizations
- List graph analytics use cases
- Application of graph analytics

Text books:

1. Data Science and Big Data Analytics-Discovering Analyzing ,Visualizing and Presenting Data –EMC Education Services, Wiley Publications, 2015**.** (Unit - I, II)
2. Big Data Analytics-from Strategic Planning to Enterprise Integration with tool, Techniques, NOSQL, and Graph BY David Loshin, Elsever Publications-2013 (Unit- III, IV and UNIT-V)

Course Outcomes:

- Able to know the introductory concepts of big data and data analytics life cycle
- Illustrate Graphical User Interface and concepts in R.
- Understand HDFS, Map Reduce and Yarn
- Able to build an evaluation model
- Able to learn NOSQL concepts

Prepared by: Dr. M. Sesha Shayee Verified by : Prof. V. NagaLakshmi

M.C.A.- V SEMESTER
GENERIC ELECTIVE - II
SCA845 J2EE TECHNOLOGIES

Hours per week: 4 End Examination: 60 Marks
Credits: 4 Sessionals : 40 Marks

Preamble:

J2EE is used for the development of web based applications with the help of html pages or Java applets and applications. J2EE combines Java technology*-based applications or services to create highly customized applications or services. The main aim of this course is to provide scalable modular application assembly and portable deployment of J2EE applications into any J2EE product through JSP and Servlets.*

Course Objectives:

- To understand the importance of various advanced java features like jdbc, XML, java servlets, jsp.
- To acquainting the student with a framework like Spring3.0. T
- To make the students to develop a web application.

UNIT - I

JDBC: ODBC – Definition, Features, Components, Architecture, JDBC- Features, Components, Architecture, JDBC API-introduction, Types , Major Classes and Interfaces, Communicating with Databases by using JDBC APIs, Creating Simple Application, Exploring the Methods of Statement, Using Statement, Working with the Prepared Statement and Callable Statement, Result Set, Result Set Meta Data interface, Advanced Concepts in Result Set, Closing the Connection.

XML: Basics of XML, Well-Formed XML, XML Namespaces, Defining Namespaces, Document Type Definitions, XML Schemas, XSLT. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Definition of JDBC, ODBC, components and architecture
- List major classes and Interfaces
- Demonstrating the communication with databases
- Working with Prepared Statements and Callable statements
- Creating a simple application

UNIT - II

Java Servlet: Introduction, Exploring the Servlet Container, Servlet API, Implementing a Servlet Object, Servlet Life Cycle, Developing First Servlet Application, Understanding Request Processing Workflow, Working with Generic Servlet Class, Implementing Servlet Request Interface, Working with Initialization Parameters, Context Initialization Parameters, Understanding Servlet Response, Understanding Request Dispatch, Dispatching the Request, Describing Request Attributes, Http Servlet Request and Http Servlet Response Interfaces, Http Servlet Life Cycle.

Handling Sessions in Servlet: Session Tracking, Describing URL rewriting, Exploring and working Hidden Form Field, Describing Cookies, Using Cookies, Exploring and Working HttpSession, Describing Servlet Context Attributes.
Filters: Filter API, working with Filters, Describing Listeners and Wrappers in Servlets, Applet to Servlet Communication. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define servlet lifecycle, servlet API
- Understanding Request Processing Workflow
- Applying Generic Servlet Class
- Understanding Request Dispatch, t, Describing Request Attributes
- Developing First Servlet Application

UNIT - III

JSP: Introduction, Advantages, JSP Life Cycle, Types of Scripting Tags, Implicit Objects, Directive Tags, Java Bean, Action Tags, Declaring a Bean in a JSP, Custom Tags- Need, Elements, Empty Tags, Body Content Tag Interface, Iteration Tag Interface, Simple Tags. (10)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define JSP life cycle
- List Types of Scripting Tags, Implicit Objects, Directive Tags
- Declaring a Bean in a JSP, Custom Tags- Need, Elements, Empty Tags

UNIT - IV

JSP Expression Language: Syntax, Types of EL Expressions, Resolving EL Expressions, EL Operators, Implicit EL Objects.

Implementing Internationalization: Introduction, Java and Internationalization, Internationalizing Web Applications.

JSTL: Describing JSTL Core, SQL, Formatting, XML tags. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define Syntax
- List Types of EL Expressions
- Describe Describing JSTL Core
- Understanding Implementing Internationalization.
- Comparing Java and Internationalization
- Creating web applications

UNIT - V

Spring 3.0: Introduction, Spring Bean- Bean Life Cycle, Bean Scopes, Inner Bean, Collection Bean, Bean Reference, Spring Auto Wiring- by Type, Name, Constructor, AutoDetect, Spring IOC tutorial – Spring DI using Setter Method (No Arg, Constructor, No ArgFactory Method),

Spring DI using Constructor, Spring AOP, Spring Annotation, Spring Validation, Example, Spring transaction Management, Spring MVC. (8)

Learning Outcomes:

By the end of this Unit, the student will be able to

- Define Spring Bean- Bean Life Cycle, Bean Scope
- Understand Inner Bean, Collection Bean, Bean Reference
- Understand AutoDetect, ArgConstructor, spring DI using constructor
- Understand Spring Annotation, Spring transaction Management and Spring MVC

Text Books:

1. JDBC, Servlet, and JSP includes JSF and Design Patterns by Santosh Kumar K, DreamTech Press, New edition, 2013.
2. Beginning XML by Liam R. E. Quin, Joe Fawcett, Danny Ayers, Wiley, 5th editon, 2012.
3. Web reference - http://javabeginnerstutorial.com/spring.

Reference Books:

1. Head First Servlets and JSP by Bryan Basham, Kathy Sierra and Bert Bates, reilly, 2nd edition, 2008.
2. Spring in Action by Craig Walss, Manning publications, 4th edition, 2014.
3. Beginning XML by David Hunter, Jeff Rafter , Andrew Watt , Joe Fawcett , Jon Duckett, Daniel J. Ayers, Wiley, 4th edition,2007.

Course Outcomes:

- Able to learn various advanced java features like jdbc, XML, java servlets and jsp.
- Able to get indepth knowledge of Servlet Object, container and application
- Understand Request Processing Workflow, Request Dispatch, Describing Request Attributes
- Illustrate Types of Scripting Tags, Implicit Objects, Directive Tags and custom tags
- Able to know Bean Life Cycle, Bean Scope, Inner Bean, Collection Bean, Bean Reference

Prepared by: Dr. M. Sesha Shayee Verified by : Prof. V. NagaLakshmi

M.C.A.- V SEMESTER
GENERIC ELECTIVE - II
SCA 881 DevOps Lab

Hours per week: 4 Examination: 100 Marks

Credits: 2

1. Understand the environment for Devops.
2. Install the required packages of AWS SDK.
3. Create a RESTful Design.
4. Understand Linux Container.
5. Install the Docker for Ubunut/CentOS.
6. Use Docker history, Docker diff, Docker create, Docker inspect.
7. Use Docker pull, Docker Registry and the commands relate to Docker.
8. Connect to containers, work with Docker file.
9. Understand Kubertenes.
10. Install and run minikube.
11. Use kubectl.
12. Setup Kubernets on AWS.
13. Setup Kubernets on Google Cloud.
14. Launch Realtime AppEngine on Google Cloud – via Kubernetes.
15. Install Kubernetes Master on Ubuntu.
16. Install Kubernetes Node 1 & Node2 on Ubuntu.
17. Connect Node1 & Node2 to Master Node.
18. Create POD Networking by using Flannel.
19. Create Kubernetes job via YAML File/Describe job/ Check Logs of K8s job.
20. Deploy a POD.
21. Manual Scaling Applications in K8s.
22. Demonstrate Self-Healing (How Ks maintains Desired State).

Reference Book:

1. DevOps with Kubernetes: Accelerating software delivery with container by Hideto Saito, Hui-Chuan Chloe Lee, Cheng-Yang Wu, O' Reilly publications, 2017.
2. Managing Kubernetes:Operating Kubernetes Clusters in the Real Worls by Brendan Burns, Craig Tracey, O'Reilly publications, 2017.

Course Outcomes:

- Able to understands environment for DevOps
- Creates a RESTful Design
- Able to learn how to work with Docker file
- Demonstrate the ctl on Kubernetes
- Practice how to Setup Kubernetes on AWS and Google Cloud
- Able to deploy a PoD

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.- V SEMESTER
GENERIC ELECTIVE - II
SCA 883 BIG DATA LAB

Hours per week: 4 Examination:100Marks

Credits: 2

1. VPN client installation on windows.
2. HDFS Commands-mkdir, put, cat, get, rm, copyFromLocal, copyToLocal, moveFromLocal, getmerge, setrep etc.
3. Linux commands-pwd,ls, cat,cal,cp,mv rm,more,less etc.
4. How to install Hadoop-read data and write data.
5. Developing a Map-reduce application.
6. MapReduce type and format. MapReduce features.
7. Implement word count / frequency programs using MapReduce.
8. Hadoop Operations-setting up a cluster, Administration of Hadoop.
9. Implement an application that stores big data in Hbase / flume / scoop/Pig/hive/spark.
10. Example problems in Hadoop Pig and its Architecture.
11. Installation and running Zoo Keeper with examples.
12. Visualize data using any plotting framework Eg. ggplot using R.

Reference Books:

1. Hadoop: The Definitive Guide – Storage and Analysis at Internet by Tom White, O'Reilley Press, 2nd edition, 2013.
2. HBase: The Definitive Guide by Lars George, O'Reilley, 2015.
3. Programming Pig – Dataflow scripting with Hadoop by Alan Gates and Daniel Dai O'Reilley, 2nd Edition, 2016.
4. An Introduction to Statistical Learning with Applications in Rby Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani Springer Publications, 2015.

Course Outcomes:

- Able to learn how to do VPN installation on windows.
- Able to practice various HDFS commands.
- The student will develop Map-reduce application.
- It helps the student in working on Hadoop ,pig and its architecture.
- The student will able to visualize data using plotting framework

Prepared by: Dr. M. Sesha Shayee Verified by : Prof. V. NagaLakshmi

M.C.A.- V SEMESTER
GENERIC ELECTIVE - II
SCA 885 J2EE TECHNOLOGIES LAB

Hours per week: 4 Examination: 100 Marks
Credits: 2

1. Write a XML program to maintain the student database.
2. Write the XML program to implement the Internal DTD and External DTD.
3. Program on JDBC for insertion, deletion, updation of data in the database.
4. Program for Multiple Insertions, Multiple Deletions, Multiple Updations of data in the database.
5. Program on Prepared Statement and Callable Statements.
6. Program to demonstrate life cycle of a servlet.
7. Program to demonstrate handling of request parameters using servlets.
8. Program to demonstrate handling of init parameters using servlets.
9. Program to create a login page using HTML &check Database values in Servlet using JDBC.
10. Program to demonstrate include() and forward() methods of request dispatcher.
11. Program to create a Filter in Servlets.
12. Program to create a session in Servlets.
13. Program to demonstrate cookies in servlets.
14. Program to demonstrate scripting tags.
15. Program to demonstrate implicit objects.
16. Program to create login page &check values of Database in JSP using JDBC, handle exception.
17. Program to demonstrate jsp:include, jsp:forward action tags.
18. Program to create a login page and check the values of Database in jsp using jdbc and also handle the exception.
19. Program to create a bean, also demonstrate setting and accessing the bean properties.
20. Program to demonstrate custom tags.
21. Program to demonstrate core tags.
22. Program to demonstrate sql tags.
23. Program to demonstrate internationalization tags.
24. Program to demonstrate EL operators.
25. Programs related to Spring using MVC architecture.-registration mechanism
26. Programs related to Spring using MVC architecture.-login mechanism.

Reference Books:

1. JDBC, Servlet and JSP includes JSF and Design Patterns Black Book by Santhosh Kumar K., New Edition, Dreamtech Press, 2013.
2. Web Technology and Design by C. Xavier, New Age International Publishers, 2003.
3. JSP and Servlets - A comprehensive Study by Mahesh P. Matha, PHI, 2013

Course Outcomes:

- Able to develop web based applications.
- Able to build web interfaces with XML,JSPs and Servlets
- Able to make servlets cooperate and share data
- Able to create jsp pages using built in jsp objects
- Able to create web pages using directives, using scripting elements, java beans and custom tags.

Prepared by: Dr. M. Sesha Shayee Verified by : Prof. V. NagaLakshmi

M.C.A.-V SEMESTER
SCA 887 MICROSOFT POWER BI LAB

Hours per week: 3 End Examination: 100 Marks

Credits: 2

1. Introduction BI Basics,
2. Power BI Desktop Overview
3. Data Discovery with Power BI Desktop
4. Transforming Data –
    - Basic Transforms,
    - Add Column From Example,
    - Appending Queries, Merging Queries
    - Combine Files, M Query Basics
    - Parameters and Templates
    - Other Query Features.
5. Introduction to Modeling Data
    - Creating the Data Model-Modeling Basics, Model Enhancements, What If Parameters.
    - Creating Calculated Columns and Tables-DAX Basics, Navigation Function, Calculated Tables.
    - Creating Calculated Measures -Measure Basics, Time Intelligence Functions.
6. Introduction to Visualizing Data
    - Creating Basic Reports with the Power BI Desktop
    - Creating Interactive Reports -Adding Slicers for Filters, Visualizing Tabular Data, Visualizing Tabular Data, Visualizing Data Trends, Visualizing Categorical and Trend Data Together, Visualizing Geographical Data with Maps, Visualizing Goal Tracking, Using Custom Visuals, Digital Storytelling, other features.
    - Using the Power BI Service -Deploying to the Power BI Service, Creating and Sharing Dashboards, Setting up App Workspaces, Subscriptions and Alerts, Excel Integration, Export and Embed Options
7. Refreshing the Data-Refreshing Data Overview, Installing the Data Gateway, Scheduling a Data Refresh
8. Power BI Report Server
    - Deployment -Publish to Report Server, Uploading from Report Server, Folder Structure

Text Books:

1.Introducing Microsoft Power BI: Alberto Ferrari and Marco Russo, Microsoft Press 2016

2.Beginning Big Data with Power BI and Excel 2013: Neil Dunlop, A press 2015

Course Outcomes:

- Able to learn introductory concepts of power BI
- Able to understand how to transform and model the data
- Design Interactive reports
- Will be able to Deploy Power BI services
- Practice the features of Power BI

Prepared by: Prof. V. NagaLakshmi Verified by : Prof. V. NagaLakshmi

M.C.A.-V SEMESTER
SCA 891 SUMMER INTERNSHIP

Credits: 2 End Examination: 100 Marks

M.C.A.-VI SEMESTER
SCA 892 MAJOR PROJECT

Credits: 8 Examination: 200 Marks