CHOICE BASED CREDIT SYSTEM

Semester Scheme with Multiple Entry and Exit Options for

Four Year Undergraduate Programme

COURSE STRUCTURE
&
DETAILED SYLLABUS
For
BACHELOR OF COMPUTER SCIENCE

MANIPUR UNIVERSITY

CANCHIPUR, IMPHAL

About the Course:

The 4-Year Undergraduate Programme (Computer Science) is designed to develop analytical & computationalthinking, and problem solving skills. It covers the core computer science topics like computersystems architecture, data structures, computer networks, operating systems, computer graphics, algorithms, software engineering, database management, theory of computation, artificialintelligence, and information security. The programme builds a base for entry level jobs ininformation technology and

Course Structure for the 4-Year Undergraduate Programme (Computer Science)							
Semester	Core (Credit)	DSE (Credit)	GEC (Credit)	AECC (Credit)	SEC (Credit)	VAC (Credit)	Semester Credit
I	CSC101(6) CSC102(6)			AECC-1 (4) English/MIL	CSSE-101(4)	VAC-101(2) VAC-102(2)	24
II	CSC203(6) CSC204(6)			AECC-2 (4) Environmental Sc.	CSSE-202(4)	VAC-203(2) VAC-204(2)	24
Exit Option with Bachelor's Certificate in Computer Science on Completion of courses equal to a minimum							

prepares the students for higher studies in the area of ComputerScience/Applications.

of 46 Credits							
III	CSC305(6) CSC306(6) CSC307(6)		CSGE301(6)			VAC-305(2)	26
IV	CSC408(6) CSC409(6) CSC410(6)		CSGE402(6)			VAC-406(2)	26
Exit Option with Bachelor's Diploma in Computer Science on Completion of courses equal to a minimum of 96 Credits							
V	CSC511(6) CSC512(6)	CSE-501(6)	CSGE-503(6)			VAC-507(2)	26
VI	CSC613(6) CSC614(6)	CSE-602(6)	CSGE-604(6)			VAC-608(2)	26
Exit Option with Bachelor's Degree in Computer Science on Completion of courses equal to a minimum of 140 Credits							
VII	CSC715(6) CSC716(6)	CSE-703(6)	CSGE-705(6)				24
VIII	CSC817(6) CSC818(6)	CSE-804(6)	CSGE-806(6)				24
Award of Bachelor's Degree with Honours in Computer Science on Completion of courses equal to a minimum of 182 Credits							

SEMESTER-WISE DISTRIBUTION OF COURSES

A. Discipline Specific Core(DSC) Courses

All the courses have 6 credits with 4 credits of theory (4 hours per week) and 2 credits of practical (4 hours per week) Or, 5 credits of theory (5 hours per week) and 1(one) credit of tutorial (1 hour per week).

SL. No.	DSC Paper Code	Semester	Course Name
1	CSC101	I	Programming Fundamentals using C
2	CSC102	I	Computer System Architecture
3	CSC203	II	Programming in Java
4	CSC204	II	Discrete Structure
5	CSC305	III	Data Structure
6	CSC306	III	Operating System
7	CSC307	III	Computer Networks
8	CSC408	IV	Design and Analysis of Algorithms
9	CSC409	IV	Internet Technologies
10	CSC410	IV	Database Management System
11	CSC511	V	Theory of Computation
12	CSC512	V	Probability and Statistics
13	CSC613	VI	Artificial Intelligence
14	CSC614	VI	Computer Graphics
15	CSC715	VII	Software Engineering
16	CSC716	VII	Data Mining
17	CSC817	VIII	Information Security
18	CSC818	VIII	Digital Image Processing

B. Discipline Specific Electives(DSE)

All the courses have 6 credits with 4 credits of theory (4 hours per week) and 2 credits of practical (4 hours per week) Or, 5 credits of theory (5 hours per week) and 1(one) credit of tutorial (1 hour per week).

DSE-1(5^{th} Semester) (Choose any one)

a) CSE501A Data Analysis and Visualization
b) CSE501B Operation Research for Computer Science
c) CSE501C Computer Oriented Numerical Methods

DSE-2(6^{th} Semester) (Choose any one)

a) CSE602A System Programming
b) CSE602B Microprocessor
c) CSE602C Modelling and Simulation

DSE-3(7^{th} Semester) (Choose any one)

a) CSE703A Advance Algorithm
b) CSE703B Combinatorial Optimization
c) CSE703C Introduction to Data Sciences
d) CSE703D Machine Learning

DSE-4(8^{th} Semester) (Choose any one)

a) CSE804A Deep Learning
b) CSE804B Unix Network Programming
c) CSE804C Project Work/Dissertation

C. Skill Enhancement Course(SEC)

All courses have 4 credits each with 2 credits of theory and 2 credits of Practical

SEC-1(First Semester): (Choose any one)

a) CSSE101A Web Design and Development
b) CSSE101B UNIX/LINUX Programming
c) CSSE101C Office Automation Tools

SEC-2(Second Semester): (Choose any one)

a) CSSE202A Programming in Python
b) CSSE202B Introduction to R programming

D. Ability Enhancement Compulsory Courses

All the courses have 4 credits each.

SL. NO.	AECC Paper Code	Semester	AECC Name
1.	AECC-1	I	English/MIL
2.	AECC-2	II	Environmental Science

E. Value Addition Courses

SL. No.	VAC Paper Code	Semester	VAC Name
1	VAC-101	I	Yoga
2	VAC-102	I	Sports
3	VAC-203	II	Culture
4	VAC-204	II	Health Care
5	VAC-305	III	NCC
6	VAC-406	IV	Ethics
7	VAC-507	V	NSS
8	VAC-608	VI	History of Science

F. Generic Elective Courses

All the courses have 6 credits with 4 credits of theory(4 hours per week) and 2 credits of practical (4 hours per week) Or, 5 credits of theory(5 hours per week) and 1(one) credit of tutorial(1 hour per week).

These courses are meant for students of other departments/disciplines.

SL. No.	GEC Paper Code	Semester	GEC Name
1	CSGE301	III	Programming Fundamentals using C
2	CSGE402	IV	Programming in Java
3	CSGE503	V	Computer System Architecture
4	CSGE604	VI	Design and Analysis of Algorithms
5	CSGE705	VII	Data Structures
6	CSGE806	VIII	Database Management System

Contents of Courses for BSc degree with honours in Computer Science

Semester	Course Code		Credit	Paper Title	Marks		Remarks
					S.A.	I.A.	
I	CSC101	Theory	4	Programming Fundamentals using C	75	25	
	CSC101P	Practical	2	-do-	35	15	
	CSC102	Theory	4	Computer System Architecture	75	25	
	CSC102P	Practical	2	-do-	35	15	
	CSSE101	Theory	2	Web Design and Development or, UNIX/LINUX Programming or, Office Automation Tools	35	15	
	CSSE101P	Practical	2	-do-	35	15	
II	CSC203	Theory	4	Programming in Java	75	25	
	CSC203P	Practical	2	-do-	35	15	
	CSC204	Theory	5	Discrite Structure	75	25	
	CSC204T	Tutorial	1	-do-	-	-	
	CSSE202	Theory	2	Programming in Python or, Introduction to R programming	35	15	
	CSSE202P	Practical	2	-do-	35	15	
Exit Option with Certificate							
III	CSC305	Theory	4	Data Structures	75	25	
	CSC305P	Practical	2	-do-	35	15	
	CSC306	Theory	4	Operating Systems	75	25	
	CSC306P	Practical	2	-do-	35	15	
	CSC307	Theory	4	Computer Networks	75	25	
	CSC307P	Practical	2	-do-	35	15	
	CSGE301	Theory	4	Programming Fundamentals using C	75	25	
	CSGE301P	Practical	2	-do-	35	15	
IV	CSC408	Theory	4	Design and Analysis of Algorithms	75	25	
	CSC408P	Practical	2	-do-	35	15	
	CSC409	Theory	4	Internet Technologies	75	25	
	CSC409P	Practical	2	-do-	35	15	
	CSC410	Theory	4	Database Management System	75	25	
	CSC410P	Practical	2	-do-	35	15	
	CSGE402	Theory	4	Programming in Java	75	25	
	CSGE402P	Practical	2	-do-	35	15	
Exit Option with Diploma							
V	CSC511	Theory	5	Theory of Computation	75	25	
	CSC511T	Tutorial	1	-do-	-	-	
	CSC512	Theory	5	Probability and Statistics	75	25	
	CSC512T	Tutorial	1	-do-	-	-	
	CSE501	Theory	4/5	Data Analysis and Visualization or, Operation Research for Computer Science (no practical) or, Computer Oriented Numerical Methods	75	25	
	CSE501P CSE501T	Practical/ Tutorial	2 1	-do-	35 -	15 -	

	CSGE503	Theory		Computer System Architecture	75	25	
	CSGE503P	Practical		-do-	35	15	
VI	CSC613	Theory		Artificial Intelligence	75	25	
	CSC613P	Practical		-do-	35	15	
	CSC614	Theory		Computer Graphics	75	25	
	CSC614P	Practical		-do-	35	15	
	CSE602	Theory		System Programming Or, Microprocessor Or, Modelling and Simulation	75	25	
	CSE602P	Practical		-do-	35	15	
	CSGE604	Theory		Design and Analysis of Algorithms	75	25	
	CSGE604P	Practical		-do-	35	15	
Exit Option with Bachelor of Science Degree							
VII	CSC715	Theory		Software Engineering	75	25	
	CSC715P	Practical		-do-	35	15	
	CSC716	Theory		Data Mining	75	25	
	CSC716P	Practical		-do-	35	15	
	CSE703	Theory		Advanced Algorithm(no practical) or, Introduction to Data Science or, Combinatorial Optimization(no practical) or, Machine Learning	75	25	
	CSE703P CSE703T	Practical Tutorial		-do-	35	15	
	CSGE705	Theory		Data Structures	75	25	
	CSGE705P	Practical		-do-	35	15	
VIII	CSC817	Theory		Information Security	75	25	
	CSC817P	Practical		-do-	35	15	
	CSC818	Theory		Digital Image Processing	75	25	
	CSC818P	Practical		-do-	35	15	
	CSE804	Theory	4/6	Deep Learning Or, Unix Network Programming Or, Project Work/Dissertation	75	25	
	CSE804P CSE804T	Practical Tutorial	2	-do-	35 -	15 -	
	CSGE806	Theory	4	Database Management System	75	25	
	CSGE806P	Practical	2	-do-	35	15	
Award of Bachelor’s Degree with Honours in Computer Science							

Courses for Programme B.Sc. Computer Science

Semester - I

CSC101: Programming Fundamentals using C
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objectives

1. To express algorithms and draw flowcharts in a language independent manner.
2. To teach how to write modular, efficient and readable C programs
3. To impart knowledge in creating and using Arrays of the C data types.
4. To describe the techniques for creating program modules in C using functions and recursive functions.
5. To demonstrate creation of derived data types and perform operations on files.
6. To understand pointers and dynamic memory allocation functions for efficiently solving problems.

Course Learning Outcomes: Upon completion of the course, the students will be able to:

1. Write, compile and debug programs in C language.
2. Use different data types in a computer program.
3. Design programs involving structures, loops, arrays and functions.
4. Identify the difference between call by value and call by reference
5. Use pointers to understand the dynamics of memory
6. Create and perform different file operations.

Detailed Syllabus

UNIT-I: **15**

Introduction to the C Language – Algorithm, Pseudo code, Flow chart, Background, C Programs, Identifiers, Data Types, Variables, Constants, Input / Output, Operators(Arithmetic, relational, logical, bitwise etc.), Expressions, Precedence and Associatively, Expression Evaluation, Type conversions. C Pre-processor - File inclusion, Macro substitution.

UNIT-II: **10**

Statement**s**- Selection Statements(making decisions) – if and switch statements, Repetition statements(loops)-while, for, do-while statements, Loop examples, other statements related to looping – break, continue, go to, Simple C Program examples.

UNIT-III: **15**

Function**s**- Introduction to Structured Programming, Functions- basics, user defined functions, inter function communication(call by value, call by reference), Standard functions. Storage classes-auto, register, static, extern; scope rules, recursive functions, example C programs.

UNIT–IV: **20**

Arrays– Basic concepts, one-dimensional arrays, two – dimensional arrays, multidimensional arrays, C programming examples Pointers – Introduction (Basic Concepts), pointers to pointers, compatibility, Pointer Applications, Arrays and Pointers, Pointer Arithmetic, memory allocation functions, array of pointers, pointers to void, pointers to functions, command–line arguments, Introduction to structures and unions.

UNIT-V: **15**

Strings – Concepts, C Strings, String Input / Output functions, string manipulation functions, string /data conversion. Input and Output – Concept of a file, streams, text files and binary files, Differences between text and binary files, State of a file, Opening and Closing files, file input / output functions (standard library input / output functions for files), file status functions (error handling), Positioning functions.

CSC101 P: Programming Fundamentals using C(Practical)

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series, S =1+1/2+1/3+1/4+……
4. WAP to compute the sum of the first n terms of the following series, S =1-2+3- 4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
*******
*********
```

10. WAP to perform following actions on an array entered by the user:
    i) Print the even-valued elements
    ii) Print the odd-valued elements
    iii) Calculate and print the sum and average of the elements of array
    iv) Print the maximum and minimum element of array
    v) Remove the duplicates from the array
    vi) Print the array in reverse order

    The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
16. Write a menu driven program to perform following operations on strings: a) Show address of each character in string b) Concatenate two strings without using strcat function. c) Concatenate two strings using strcat function. d) Compare two strings e) Calculate length of the string (use pointers) f) Convert all lowercase characters to uppercase g) Convert all uppercase characters to lowercase h) Calculate number of vowels i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration.
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration.
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.

21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order. These are only examples, more can be included related to the theory
25. WAP to sort the elements of any array.
26. WAP to search an array for an element.

CSC102: Computer System Architecture
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces the students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system.

Course Learning Outcomes

On successful completion of the course, students will be able to:
1. Design Combinational Circuits using basic building blocks. Simplify these circuits usingBoolean algebra and Karnaugh maps. Differentiate between combinational circuits andsequential circuits.
2. Represent data in binary form, convert numeric data between different number systems andperform arithmetic operations in binary.
3. Determine various stages of instruction cycle and describe interrupts and their handling.
4. Explain how CPU communicates with memory and I/O devices.
5. Simulate the design of a basic computer using a software tool

Detailed Syllabus

UNIT-I: 10

Digital Logic Circuits:Logic Gates, truth tables, Boolean Algebra, digital circuits,combinational circuits, sequential circuits, circuit simplification using Karnaugh map, Don'tCare Conditions, flip-flops, characteristic tables

UNIT-II: 10

Digital Components: Half Adder, Full Adder, Decoders, Multiplexers, Registers and Memory Units

UNIT-III: 10

Data Representation and Basic Computer Arithmetic: Number system, complements, fixed and floating point representation. Alphanumeric representation. Addition, subtraction. **UNIT-IV: 15**

Basic Computer Organization and Design: Common Bus system, instruction codes, instruction format, instruction set completeness, Sequence Counter, timing and control, instruction cycle, memory reference instructions and their implementation using arithmetic, logical, program control, transfer and input output micro-operations, interrupt cycle. **UNIT-V: 15**

Central Processing Unit: Micro programmed Control vs Hardwired Control, lower level programming languages, Instruction format, accumulator, general register organization, stack organization, zero-address instructions, one-address instructions, two-address instructions, threeaddress instructions, Addressing Modes, RISC, CISC architectures, pipelining and parallel

processing.

UNIT-VI: **15**

Memory Organization and Input-Output Organization: Input-Output Organization: Peripheral Devices, I/O interface, I/O vs. Memory Bus, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access

CSC102 P: Computer System Architecture (Practical)

(Use Simulator – CPU Sim 3.6.9 or any higher version for the implementation)

1. Create a machine based on the following architecture:

Registers						
IR	DR	AC	AR	PC	I	E
0 15	0 15	0 15	0 11	0 11	1 bit	1 Bit

Memory 4096 words 8 bits per word	Instruction format 0 3 4 15 Opcode | Address

Basic Computer Instructions

Memory Reference			Register Reference		
Symbol	Hex		Symbol	Hex	
AND	0xxx	Direct Addressing	CLA	E800	
ADD	2xxx		CLE	E400	
LDA	4xxx		CMA	E200	
STA	6xxx		CME	E100	
BUN	8xxx		CIR	E080	
			CIL	E040	
ISZ	Cxxx		INC	E020	
AND_I	1xxx	Indirect Addressing	SPA	E010	
ADD_I	3xxx		SNA	E008	
LDA_I	5xxx		SZA	E004	
STA_I	7xxx		SZE	E002	
BUN_I	9xxx		HLT	E001	
ISZ_I	Dxxx				

Refer to Chapter-5 of reference 1 for description of instructions.

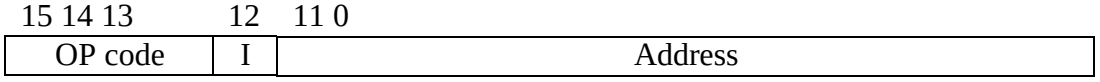
Design the register set, memory and the instruction set. Use this machine for the assignments

of this section.

2. Create a Fetch routine of the instruction cycle.
3. Write an assembly program to simulate ADD operation on two user-entered numbers.
4. Write an assembly program to simulate SUBTRACT operation on two user-entered numbers.
5. Write an assembly program to simulate the following logical operations on two userentered numbers.
    1. AND
    2. OR
    3. NOT
    4. XOR
    5. NOR
    6. NAND
6. Write an assembly program to simulate MULTIPLY operation on two user-entered numbers.
7. Write an assembly program for simulating following memory-reference instructions.
    1. ADD
    2. LDA
    3. STA
    4. BUN
    5. ISZ
8. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
    1. CLA
    2. CMA
    3. CME
    4. HLT

9. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
    1. INC
    2. SPA
    3. SNA
    4. SZE
10. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
    1. CIR
    2. CIL
11. Write an assembly program that reads in integers and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).
12. Write an assembly program that reads in integers and adds them together; until zero is read in. Then it outputs the sum.
13. Create a machine for the following instruction format:

Instruction format



The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address.

Write an assembly program to simulate the machine for addition of two numbers with I= 0 (Direct Address) and address part = 082. The instruction to be stored at address 022 in RAM, initialize the memory word with any decimal value at address 082. Determine the contents of AC, DR, PC, AR and IR in decimal after the execution.

14. Simulate the machine for the memory-reference instruction referred in above question with I= 1 (Indirect Address) and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand 632 and AC with 937. Determine the contents of AC, DR, PC, AR and IR in decimal after the execution.

15. The instruction format contains 3 bits of opcode, 12 bits for address and 1 bit for addressing mode. There are only two addressing modes, I = 0 is direct addressing and I = 1 is indirect addressing. Write an assembly program to check the I bit to determine the addressing mode and then jump accordingly.

References

1. Mano, M. (1992). *Computer System Architecture*. 3rd edition. Pearson Education. **Additional Resources**1. Mano, M. (1995). *Digital Design*. Pearson Education Asia. **2.** Null, L., & Lobur, J. (2018). *The Essentials of Computer Organization and Architecture*. 5th edition. (Reprint) Jones and Bartlett Learning.3. Stallings, W. (2010). *Computer Organization and Architecture Designing for Performance* 8thedition. Prentice Hall of India.**Course Teaching Learning Process**

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Tentative weekly teaching plan is as follows:

Week	Content
1 – 2	Unit 1 - Introduction:Digital Logic Gates, Flipflops and their characterstic table, Logic circuit simplification using Boolean Algebra and Karnaugh Map,Don't Care conditions. Combinational Circuits, Sequential Circuits.
3 – 4	Unit 2 - Digital Components:Decoders, Encoders, Multiplexers, Binary Adder, Binary Adder Subtractor, Binary Incrementer, Registers and Memory Units
5 – 6	Unit 3 - Data Representation:Binary representation of data, representation of alpha data,representation of numeric data in different number systems, conversion between number systems, complements, representation ofdecimal numbers, representation of signed and unsigned numbers, addition and subtraction of signed and unsigned numbers and overflow detection.
7 – 11	Unit 4 - Operations and Control:Arithmetic and logical micro-operations, micro programmed controlvs. hardwired control, instruction format, instruction setcompleteness, timing and control, instruction cycle, memoryreference instructions and their implementation using arithmetic,logical, program control, transfer and input output micro operations,interrupt cycle.

12 - 13	Unit 5 - Instructions:Instruction format illustration using single accumulator organization, general register organization and stack organization, AddressingModes, zero-address instructions, one-address instructions, twoaddress instructions and three-address instructions,
14 - 15	Unit 6 - Peripheral Devices:I/O interface, I/O vs. Memory Bus, Isolated I/O, Memory MappedI/O, Direct Memory Access

Semester - II

CSC203:Programming in JAVA
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course adds to the basic programming language skills acquired by the student in earliersemesters. The students are exposed to the advanced features available in Java such as exceptionhandling, file handling, interfaces, packages and GUI programming.

Course Learning Outcomes

On successful completion of the course the student will be

1. Implement Exception Handling and File Handling.
2. Implement multiple inheritance using Interfaces.
3. Logically organize classes and interfaces using packages.
4. Use AWT and Swing to design GUI applications.

Detailed Syllabus

UNIT-I: **15**

Introduction to Object Oriented Approach: History and evolution of Object Oriented Languages. OOPs features: Encapsulation, Data Abstraction, Inheritance, Multiple Inheritance, Polymorphism, Message Passing, Extensibility, Persistence, Delegation, Genericity.

UNIT-II: **10**

Introduction to Java: Features of Java – Object Oriented Concepts – Lexical Issues – Data Types – Variables – Arrays – Operators – Control Statements.

UNIT-III: **10**

Classes, Objects – Constructors – Overloading method – Access Control – Static and fixed methods – Inner Classes – String Class – Inheritance – Overriding methods – Using super – Abstract class.

UNIT-IV: **10**

Interfaces Interface basics; Defining, implementing and extending interfaces; Implementing multiple inheritance using interfaces Packages Basics of packages, Creating and accessing packages, System packages, Creating user defined packages

UNIT-V: **10**

Exception handling using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, multiple catch statements, creating user defined exceptions.

UNIT-VI: **10**

File Handling Byte Stream, Character Stream, File I/O Basics, File Operations

UNIT-VII: 10

AWT and Event Handling: The AWT class hierarchy, Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Creating GUI applications using AWT.

CSC203 P: Programming in JAVA(Practical)

1. Design a class Complex having a real part (x) and an imaginary part (y). Provide methodsto perform the following on complex numbers:
    1. Add two complex numbers.
    2. Multiply two complex numbers.
    3. toString() method to display complex numbers in the form: x + i y
2. Create a class TwoDim which contains private members as x and y coordinates inpackage P1. Define the default constructor, a parameterized constructor and overridetoString() method to display the co-ordinates. Now reuse this class and in package P2create another class ThreeDim, adding a new dimension as z as its private member.Define the constructors for the subclass and override toString() method in the subclassalso. Write appropriate methods to show dynamic method dispatch. The main() functionshould be in a package P.

3. Define an abstract class Shape in package P1. Inherit two more classes: Rectangle inpackage P2 and Circle in package P3. Write a program to ask the user for the type ofshape and then using the concept of dynamic method dispatch, display the area of theappropriate subclass. Also write appropriate methods to read the data. The main()function should not be in any package.

4. Create an exception subclass UnderAge, which prints “Under Age” along with the agevalue when an object of UnderAge class is printed in the catch statement. Write a classxceptionDemo in which the method test() throws UnderAge exception if the variable agepassed to it as argument is less than 18. Write main() method also to show working of theprogram.
5. Write a program to implement stack. Use exception handling to manage underflow andoverflow conditions.
6. Write a program that copies content of one file to another. Pass the names of the filesthrough command-line arguments.
7. Write a program to read a file and display only those lines that have the first twocharacters as '//' (Use try with resources).
8. Write a program to create an Applet. Create a frame as a child of applet. Implement mouseClicked( ), mouseEntered( ) and mouseExited( ) events for applet. Frame is visible when mouse enters applet window and hidden when mouse exits from the applet window.
9. Write a program to display a string in frame window with pink color as background.
10. Write a program to create an Applet that has two buttons named “Red” and “Blue”. When a button is pressed the background color of the applet is set to the color named by the button’s label.
11. Create an applet which responds to KEY_TYPED event and updates the status window with message (“Typed character is: X”). Use adapter class for other two events.
12. Create an applet with two buttons labeled ‘A’ and ‘B’. When button ‘A’ is pressed, itdisplays your personal information (Name, Course, Roll No, College) and when button ‘B’is pressed, it displays your CGPA in previous semester.

13. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet's window.
14. Rewrite the applet programs using Swing.

References

1. Schildt, H. (2018). *Java: The Complete Reference*. 10th edition. McGraw-Hill Education.

Additional Resources:

1. Balaguruswamy E. (2014). *Programming with JAVA: A Primer*. 5th edition. India: McGraw Hill Education
2. Horstmann, C. S. (2017). *Core Java - Vol. I – Fundamentals* (Vol. 10). Pearson Education
3. Schildt, H., & Skrien, D. (2012). *Java Fundamentals - A Comprehensive Introduction*. India: McGraw Hill Education

CSC204:Discrete Structures
Credit: 06
Total Marks: 100 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks)
Workload: 5 Lectures (per week), 1 Tutorial(per Week)

Course Objective

The course aims to introduce the students to Boolean algebra, sets, relations, functions,principles of counting, and growth functions so that these concepts may be used effectively inother courses.

Course Learning Outcomes

On successful completion of the course, students will be able to:
1. Define mathematical structures (relations, functions, sequences, series, and graphs) and usethem to model real life situations.
2. Understand (trace) and construct simple mathematical proofs using logical arguments.
3. Solve class room puzzles based on counting principles.
4. Compare functions and relations with respect to their growth for large values of the input.

Detailed Syllabus

UNIT-I: 10

Introduction:Sets - finite and infinite sets, uncountable infinite sets; functions, relations,properties of binary relations, closure, partial ordering relations; counting - Pigeonhole Principle, permutation and combination; mathematical induction, Principle of Inclusion and Exclusion. **UNIT-II: 10**

Growth of Functions:asymptotic notations, summation formulas and properties, bounding summations, approximation by integrals.

UNIT-III: 20

Recurrence:recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, recursion trees, Master Theorem

UNIT-IV: 20

Graph Theory:basic terminology, models and types, multi-graphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and Circuits, planar graphs, graph coloring, Trees, basic terminology and properties of Trees, introduction to spanning trees.

UNIT-V 15

Propositional Logic: logical connectives, well-formed formulas, tautologies, equivalences, Inference Theory

References

1. Mohapatra, & Liu, C. L. (2012). *Elements of Discrete mathematics*. 4th edition. McGraw Hill Education.

2. Rosen, K. H. (2011). *Discrete Mathematics and Its Applications*. 7th edition. Tata McGraw Hill Education.

Additional Resources1. Albertson, M. O., & Hutchinson, J.P., (1988). *Discrete Mathematics with Algorithms.* JohnWiley and Sons.

2. Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (2009). *Introduction to algorithms*. 3rdedition. MIT Press.

3. Hein, J. L. (2015). *Discrete Structures, Logic, and Computability.* 4th edition. Jones andBartlett Learning.

4. Hunter, D. J. (2011). *Essentials of Discrete Mathematics.* 2nd edition. Jones and BartlettLearning

Tentative weekly teaching plan is as follows:

Week	Content
1-3	SetsFinite and infinite sets, uncountable infinite sets; functions, relations,properties of binary relations, closure, partial ordering relations,pigeonhole principle, permutation and combination, induction, inclusionexclusion
4-5	Growth of FunctionsAsymptotic notations, summation formulas and properties, summation formulas and properties (contd.), bounding summations, approx. byintegrals
6-8	RecurrencesRecurrence relations, generating functions, linear recurrence relationswith constant coefficients and their solution, recursion trees, Master'sTheorem
9-13	Graph TheoryBasic terminology, models and types, multigraphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler andHamiltonian Paths and Circuits, planar graphs, graph coloring, Trees,basic terminology and properties of Trees, introduction to spanning trees.
14-15	Prepositional LogicLogical connectives, well-formed formulas, tautologies, equivalences, inference theory

Semester - III

CSC305: Data Structures
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course aims at developing the ability to use basic data structures like array, stacks, queues,lists, trees and hash tables to solve problems. C++ is chosen as the language to understandmplementation of these data structures.

Course Learning Outcomes

At the end of the course, students will be able to:

1. Implement and empirically analyse linear and non-linear data structures like Arrays, Stacks,Queues, Lists, Trees, Heaps and Hash tables as abstract data structures. (RBT L2/3)
2. Write a program, choosing a data structure, best suited for the application at hand. (RBT L3/4)
3. Re-write a given program that uses one data structure, using a more appropriate/efficient data structure (RBT L4)
4. Write programs using recursion for simple problems. Explain the advantages and disadvantages of recursion.(RBT L2/L3)
5. Identify Ethical Dilemmas.

Detailed Syllabus

UNIT-I: 10

Arrays:single and multi-dimensional arrays, analysis of insert, delete and search operations in arrays (both linear search and binary search), implementing sparse matrices, applications of arrays to sorting: selection sort, insertion sort, bubble sort, comparison of sorting techniques via empirical studies. Introduction to Vectors.

UNIT-II: 10

Linked Lists: Singly- linked, doubly-linked and circular lists, analysis of insert, delete and search operations in all the three types, implementing sparse matrices. Introduction to Sequences.

UNIT-III: 10

Queues: Array and linked representation of queue, de-queue, comparison of the operations on queues in the two representations. Applications of queues.

UNIT-IV: 15

Stacks:Array and linked representation of stacks, comparison of the operations on stacks in the two representations, implementing multiple stacks in an array; applications of stacks: prefix, infix and postfix expressions, utility and conversion of these expressions from one to another; applications of stacks to recursion: developing recursive solutions to simple problems, advantages and limitations of recursion

UNIT-V: 20

Trees and Heaps: Introduction to tree as a data structure; binary trees, binary search trees, analysis of insert, delete, search operations, recursive and iterative traversals on binary search trees. Height-balanced trees (AVL), B trees, analysis of insert, delete, search operations on AVL and B trees.

Introduction to heap as a data structure. analysis of insert, extract-min/max and delete-min/max operations, applications to priority queues.

UNIT-VI: 10

Hash Tables: Introduction to hashing, hash tables and hashing functions -insertion, resolving collision by open addressing, deletion, searching and their analysis, properties of a good hash function.

CSC305 P: Data Structures(Practical)

1. Write a program to search an element from a list. Give user the option to perform Linear orBinary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting usingInsertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search ofa number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no.
    (i) using recursion,
    (ii) using iteration
12. (ii) WAP to display fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive and Iterative Implementation) (b) Deletion by copying (c) Deletion by Merging (d) Search a no. in BST (e) Display its preorder, postorder and inorder traversals Recursively (f) Display its preorder, postorder and inorder traversals Iteratively (g) Display its level-by-level traversals (h) Count the non-leaf nodes and leaf nodes (i) Display height of tree (j) Create a mirror image of tree (k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations

like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. WAP to implement various operations on AVL Tree.
24. WAP to implement heap operations.

References

1. Drozdek, A., (2012), *Data Structures and algorithm in C++*. 3rd edition. Cengage Learning.
2. Goodrich, M., Tamassia, R., & Mount, D., (2011). *Data Structures and Algorithms Analysis in C++*. 2nd edition. Wiley.

Additional Resources

1. Foruzan, B.A. (2012) *Computer Science: A Structured Approach Using C++*, Cengage Learning
2. Lafore, R. (2008). *Object Oriented Programming in C++*. 4th edition. SAMS Publishing.
3. Sahni, S. (2011). *Data Structures, Algorithms and applications in C++*. *2nd*Edition, Universities Press
4. Tenenbaum, A. M., Augenstein, M. J., & Langsam Y., (2009), *Data Structures Using C and C++*. 2nd edition. PHI

Tentative weekly teaching plan is as follows:

Week	**Content**
1 -2	Single and Multi-dimensional arrays, row and column major –order, static vs. dynamic data structures
3-4	Linked Lists, doubly linked list, circular lists, implementation of link list in array, using pointers, analysis of linked Lists operations, sparse matrices, sequences
5-6	Queues, storage and retrieval operations, implementation of queues,
7-8	Stacks, storage and retrieval operations, implementation of stacks, applications of stacks
9-10	Binary Trees, Recursive and iterative methods of tree traversal
11-13	AVL and B Trees
14	Heaps
15	Hash Tables

CSC306: Operating Systems
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

The course introduces the students to different types of operating systems. Operating systemmodules such as memory management, process management and file management are covered indetail.

Course Learning Outcomes

On successful completion of the course, the students will be able to:
1. Implement multiprogramming, multithreading concepts for a small operating system.
2. Create, delete, and synchronize processes for a small operating system.3. Implement simple memory management techniques.
4. Implement CPU and disk scheduling algorithms.
5. Use services of modern operating system efficiently

6. Implement a basic file system.

Detailed Syllabus

UNIT-I: **10**

Introduction:Operating systems (OS) definition, Multiprogramming and Time Sharing operating systems, real time OS, Multiprocessor operating systems, Multicore operating systems, Various computing environments.

UNIT-II: **10**

Operating System Structures:Operating Systems services, System calls and System programs, operating system architecture (Micro Kernel, client server) operating

UNIT-III: **20**

Process Management:Process concept, Operation on processes, Multi-threaded processes and models, Multicore systems, Process scheduling algorithms, Process synchronization. The Critical-section problem and deadlock characterization, deadlock handling.

UNIT-IV: **20**

Memory Management:Physical and Logical address space; Memory allocation strategies - Fixed and Variable Partitions, Paging, Segmentation, Demand Paging and virtual memory, Page Replacement algorithm.

UNIT-V: **15**

File and I/O Management: Directory structure, File access methods, Disk scheduling algorithms.

Practical

1. Write a program (using fork() and/or exec() commands) where parent and child execute:
a)same program, same code. b) same program, different code. - c) before terminating, the parent waits for the child to finish its task.
2. Write a program to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information).
3. Write a program to report behaviour of Linux kernel including information on 19 configured memory, amount of free and used memory. (memory information).
4. Write a program to print file details including owner access permissions, file access time, where file name is given as argument.
5. Write a program to copy files using system calls.
6. Write a program to implement FCFS scheduling algorithm.
7. Write a program to implement Round Robin scheduling algorithm.
8. Write a program to implement SJF scheduling algorithm.
9. Write a program to implement non-preemptive priority based scheduling algorithm.
10. Write a program to implement preemptive priority based scheduling algorithm.
11. Write a program to implement SRJF scheduling algorithm.
12. Write a program to calculate sum of n numbers using thread library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

References

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2008). *Operating Systems Concepts*. 8th edition.. John Wiley Publications.

Additional Resources

1. Dhamdhere, D. M. (2006). *Operating Systems: A Concept-based Approach*. 2nd edition. Tata McGraw-Hill Education.
2. Kernighan, B. W., & Rob Pike, R. (1984). *The Unix programming environment* (Vol. 270). Englewood Cliffs, NJ: Prentice-Hall
3. Stallings, W. (2018). *Operating Systems: Internals and Design Principles*. 9th edition. Pearson Education.
4. Tanenbaum, A. S. (2007). *Modern Operating Systems*. 3rd edition. Pearson Education.

Tentative weekly teaching plan is as follows:

Week	Content
1	Operating System, Definition and its purpose, Time sharing, Multiprogramming and Multiprocessing, Operating System Operations
2	Operating System Services, User and Operating System Interface, System Calls and its Types.
3	Operating system Design and Structure, System Programs, System Boot, Process
4	Operations on Processes, Inter process communication, Shared memory.
5	Multithreading Models, Multicore Programming, Thread Libraries
6	Process Scheduling criteria, Process Scheduling Algorithms, Multiple Processor Scheduling.
7	Process Synchronization, Critical Section Problem, Semaphores.
8	Deadlock Characterization, Methods for handling deadlocks.
9-10	Memory Allocation Strategies-Fixed and Variable partition, Swapping, Logical and Physical Address Space, Paging, Structure of Page Table and its Variations, Shared pages, Segmentation
11-12	Virtual memory, Page Replacement Algorithms, Allocation of frames, Thrashing, Working set model.
13-14	File System , File Characteristics, Access methods, Directory and Disk structure , File system structure and implementation, Directory implementation, Free space Implementation, File Allocation methods.
15	Overview of Secondary Devices, Disk Scheduling Algorithms

CSC307: Computer Networks
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course covers the concepts of data communication and computer networks. It comprises ofthe study of the standard models for the layered protocol architecture to communicate between autonomous computers in a network and also the main features and issues of communicationprotocols for different layers. Topics covered comprise of introduction to OSI and TCP/IPmodels also.

Course Learning Outcomes

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

1. Describe the hardware, software components of a network and their interrelations.
2. Compare OSI and TCP/IP network models.
3. Describe, analyze and compare different data link, network, and transport layer protocols.
4. Design/implement data link and network layer protocols in a simulated networking environment.

Detailed Syllabus

UNIT-I: **5**

Introduction:Types of computer networks, Internet, Intranet, Network topologies, Network classifications.

UNIT-II: **5**

Network Architecture Models:Layered architecture approach, OSI Reference Model, TCP/IP Reference Model.

UNIT-III: **15**

Physical Layer:Analog signal, digital signal, digital modulation techniques (ASK, PSK, QAM), encoding techniques, maximum data rate of a channel, transmission media (guided transmissionmedia, wireless transmission, satellite communication), multiplexing (frequency divisionmultiplexing, time division multiplexing, wavelength division multiplexing).

UNIT-IV: **15**

Data Link MAC Layer**:** Data link layer services, error-detection and correction techniques,error recovery protocols (stop and wait, go back n, selective repeat), multiple access protocols,(TDMA/FDP, CDMA/FDD/CSMA/CD, CSMA/CA), Datalink and MAC addressing, Ethernet,data link layer switching, point-to-point protocol.

UNIT-V: **10**

Network layer:Networks and Inter networks, virtual circuits and datagrams, addressing, sub netting, Routing- (Distance vector and link state routing), Network Layer Protocols- (ARP, IPV4, ICMP, IPV6).

UNIT-VI: **15**

Transport and Application Layer**:** Process to process Delivery- (client server paradigm,connectionless versus connection oriented service, reliable versus unreliable); User DatagramProtocols, TCP/IP protocol, Flow Control.

UNIT-VII: **10**

Protocols:FTP (File Transfer protocol), SMTP (Simple, Mail Transfer Protocol), Telnet and remote login protocol, WWW (World Wide Web), HTTP (Hyper Text Transfer protocol), Uniform Resource Locator, HTML and forms.

Practical

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n sliding window protocol.
4. Simulate and implement selective repeat sliding window protocol.
5. Simulate and implement distance vector routing algorithm

6. Simulate and implement Dijkstra algorithm for shortest path routing.

References:

1. Forouzan, B. A. (2017). *Data Communication and Networking*. McGraw-Hill Education
2. Tanenbaum, A.S. & Wethrall,D.J. (2012). *Computer Networks*. Pearson Education

Additional References

1. Kozierok, C.M. *The TCP/IP Guide*, free online resource. (2005.). Retrieved from http://www.tcpipguide.com/free/index.htm
2. Kurose, J. F., & Ross, K. W. (2017). *Computer Networking: A Top-Down Approach*. Pearson Education India
3. Stallings, W. (2017). *Data and Computer Communications.* 10th edition. Pearson Education India

Tentative weekly teaching plan is as follows:

Week	Topics to be covered
1	Introduction to Computer Networks: Network definition, types ofcomputer networks, Internet, intranet, network topologies, and networkclassifications.
2	Network Performance issues and concepts: Putting network performancein perspective, balancing network performance with key non-performancecharacteristics.
3	Performance measurements: speed, bandwidth, throughput and latency;simplex, half duplex and full duplex operation; Quality of service.
4	Network Architecture Models: Layered Approach, OSI Reference Model,TCP/IP Reference Model.
5	Network devices: hubs, switches, bridges, routers, gateways.Physical Layer: Analog signal, digital signal.
6	Physical Layer: digital modulation techniques (ASK, PSK, QAM),encoding techniques, frequency division multiplexing, time divisionmultiplexing.
7	Physical Layer: switching techniques- Circuit, packet and messageswitching, guided transmission media, wireless transmission, satellitecommunicationData Link Layer: data link layer services, framing and flow control.
8	Data Link Layer: error-detection and correction techniques error recoveryprotocols (stop and wait (for noiseless and noisy environment)).
9	Data Link Layer: error recovery protocols (go back n, selective repeat),multiple access protocols, addressing, Ethernet, data link layer switching,point-to-point protocol.
10-11	Network layer: Inter networks, virtual circuits and datagrams, addressingsub netting, Routing-distance vector and link state routing, NetworkLayer Protocols- ARP, IPV4, ICMP, IPV6.
12	Transport Layer: Process to process Delivery- client server paradigm,connectionless versus connection oriented service, reliable versusunreliable; user datagram Protocol- well known ports, user datagram.
13	Transport Layer: UDP Operation, use of UDP, TCP/IP protocol – wellknown ports, TCP Service, features.
14	Transport Layer: TCP connection establishment and release, FlowControl.Application Layer: Domain name space, Distribution of Name space, DNSin the Internet, Resolution.
15	Application Layer: WWW and HTTP, Architecture- Client server model,Uniform Resource Locator, HTTP-Transaction, HTTP operational modeland client server communication, HTTP message format.

Semester - IV

CSC408: Design and Analysis of Algorithms
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course is designed to introduce the students to design and analyse algorithms in terms ofefficiency and correctness. The course focuses on highlighting difference between variousproblem solving techniques for efficient algorithm design.

Course Learning Outcomes:

On successful completion of this course, the student will be able to:
1. Given an algorithm, identify the problem it solves.
2. Write algorithms choosing the best one or a combination of two or more of the algorithm design techniques: Iterative, divide-n-conquer, Greedy, Dynamic Programming using appropriate data structures.
3. Write proofs for correctness of algorithms.
4. Re-write a given algorithm replacing the (algorithm design) technique used with a more appropriate/efficient (algorithm design) technique.

Detailed Syllabus

UNIT-I: 30

Algorithm Design Techniques:Iterative technique: Applications to Sorting and Searching (review), their correctness and analysis. Divide and Conquer: Application to Sorting and Searching (review of binary search), merge sort, quick sort, their correctness and analysis. Dynamic Programming:Application to various problems (for reference; Weighted Interval Scheduling, Sequence Alignment, Knapsack), their correctness and analysis. Greedy Algorithms: Application to various problems, their correctness and analysis.

UNIT-II: 20

More on Sorting and Searching:Heapsort, Lower Bounds using decision trees, sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Medians & Order Statistics, complexityanalysis and their correctness.

UNIT-III: 10

Advanced Analysis Technique: Amortized analysis

UNIT-IV: 15

Graphs: Graph Algorithms - Breadth First Search, Depth First Search and its Applications.

CSC408 P :Design and Analysis of Algorithms(Practical)

1. a. Implement Insertion Sort (The program should report the number of comparisons)
   b. Implement Merge Sort(The program should report the number of comparisons)
2. Implement Heap Sort (The program should report the number of comparisons)
3. Implement Randomized Quick sort (The program should report the number of comparisons)

4. Implement Radix Sort
5. Create a Red-Black Tree and perform following operations on it: i. Insert a node ii. Delete a node iii. Search for a number & also report the color of the node containing this number.
6. Write a program to determine the LCS of two given sequences
7. Implement Breadth-First Search in a graph
8. Implement Depth-First Search in a graph
9. Write a program to determine the minimum spanning tree of a graph

For the algorithms at S.No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.

References

1. Kleinberg, J., & Tardos, E. (2013). *Algorithm Design*. 1st edition. Pearson Education India.

Additional Resources

1. Cormen, T.H., Leiserson,C.E. Rivest, R.L., & Stein, C.(2015). *Introduction to Algorithms.* 3rd edition. PHI.
2. Sarabasse & Gleder A. V. (1999). *Computer Algorithm – Introduction to Design and Analysis*. 3rd edition. Pearson Education

Tentative weekly teaching plan is as follows:

Week	Content
1	Iterative technique: Applications to Sorting and Searching (review), their correctness and analysis
2	Divide and Conquer: Application to Sorting and Searching (review of binary search), merge sort, their correctness and analysis.
3	Divide and Conquer: quick sort, its correctness and analysis.
4	Heapsort, its correctness and analysis
5	Lower Bounds using decision trees, sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, their analysis
6	Medians & Order Statistics with analysis
7-9	Graph Algorithms: Graph Representation, Breadth First Search, Depth First Search, Applications
10-11	Greedy Algorithms: Application to various problems, their correctness and analysis
12-14	Dynamic Programming: Application to various problems, their correctness and analysis
15	Amortized analysis

CSC409: Internet Technologies
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces the protocols used in Internet, its architecture, and security aspect ofInternet. Student will have an insight that how a search engine works and web crawls.

Course Learning Outcomes

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

1. Describe Internet, its architecture, services and protocol.
2. Implement a simple search engine.
3. Implement a web crawler.
4. Use javascript technologies to make a website highly responsive, more efficient and user friendly

Detailed Syllabus

UNIT-I: 15

Introduction:Network address translation, Subnet Masking, Difference between Intranet andInternet, Working of Internet, Dynamic and Static Routing, Domain Name Server , networkingtools - ipconfig, ping, netstat, traceroute

UNIT-II: 10

Introduction to Internet Protocols: HTTP, HTTPS, FTP, SMTP, IMAP, POP3, VoIP

UNIT-III: 15

Web Servers:Introduction, Working, Configuring, Hosting and Managing a Web server, Proxy Servers: Introduction, Working, Type of Proxies, setting up and managing a proxy server Client-side Technologies, Server-side Technologies and hybrid technologies

UNIT-IV: 15

Javascript, jQuery, JSON, NODE.js, BOOTSTRAP, Introduction to forums, blogging, portfolio, developing a responsive website, Combining Web Applications and Mobile Applications

UNIT-V: 10

Search Engines- components, working, optimization, Crawling, BOTS

UNIT-VI: 10

Introduction to cookies and sessions, Introduction to e-commerce websites and e-carts.

CSC409 P: Internet Technologies(Practical)

Pre-requisites for course: Programming, Computer Networks, Web-Designing (HTML, CSS, Basic JavaScript)

1. Demonstrate the use of networking tools like ping, ipconfig, netstat and traceroute.
2. Configure a web-server on a personal system.
3. Demonstrate the network monitoring of the internet traffic through any predefined tool
4. Develop an interactive website using jquery, JSON, NODE.js and BOOTSTRAP with following functionalities.
    1. Design a home page and other allied pages of the website using HTML and CSS
    2. Create a registration form and insert the data into tables at the backend. Creating an html form with content validation using JavaScript.

3. Handle HTML form using jQuery, store the data in JSON objects, pass them to another page and display it there using jQuery
4. Logging system to manage various types of accounts
5. Create pages with dynamic content fetching and display
6. Perform event handling in node.js

References

1. Bayross, I. (2013). *Web enabled commercial application development using HTML, JavaScript, DHTML and PHP*. 4th edition. BPB Publication.
2. DComer. (2018). *The Internet Book: Everything You need to know about Computer networking and how the internet works*. 5th edition. CRC Press.
3. Duckett, J.(2014*). JavaScript and JQuery: Interactive Front-End Web Development*. Wiley

Additional Resources

1. Godbole, A. S.& Kahate A (2008). *Web Technologies*. Tata McGrawHill
2. Greenlaw R. & Hepp E, (2007). *Fundamentals of Internet and WWW*. 2nd edition. Tata McGrawHill.
3. Jackson. (2008). *Web Technologies*. Pearson Education
4. Patel, B & Barik, L.B , *Internet & Web Technology* , Acme Learning Publisher.
5. Reddy, S., Aggarwal, A., Sayer, M., Totty, B., & Gourley, D. (2002). *HTTP: The Definitive Guide*. Media: O'Reilly Media Inc.
6. Young, M. L. (2007). *The Complete reference to Internet*. Tata: McGraw Hill.

Tentative weekly teaching plan is as follows:

Week	Content
1-2	Network address translation, Subnet Masking, Difference betweenIntranet and Internet, Working of Internet, Dynamic and StaticRouting, Domain Name Server, networking tools - ipconfig, ping,netstat, traceroute
3	Introduction to Internet Protocols - HTTP, HTTPS, FTP, SMTP, IMAP,POP3, VoIP
4-7	Web Servers: Working, Configuring, Hosting and Managing a WebserverProxy Servers: Working, Type of Proxies, setting up and managing aproxy server, Client-side Technologies, Server-side Technologies andhybrid technologies
8-10	Javascript, JSON jQuery
11-12	NODE.js, BOOTSTRAP
13-14	Introduction to forums, blogging, portfolio, Developing a responsivewebsite, combining Web Applications and Mobile Applications
15	Search Engines - components, working, optimization, Crawling, BOTS Introduction to cookies and sessions, e-commerce websites and e-carts

CSC410: Database Management Systems
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

The course introduces the foundations of database management systems focusing on significanceof a database, relational data model, schema creation and normalization, transaction processing,indexing, and the relevant data structures (files and B+-trees).

Course Learning Outcomes

On successful completion of the course, students will:

1. Describe major components of DBMS and their functions
2. Model an application's data requirements using conceptual modelling tools like ER diagramsand design database schemas based on the conceptual model.
3. Write queries in relational algebra / SQL
4. Normalize a given database schema to avoid data anomalies and data redundancy.
5. Describe the notions of indexes, views, constraints and transactions.

Detailed Syllabus

UNIT-I: 10

Introduction to databases:Characteristics of database approach, data models, database system architecture, data independence and data abstraction.

UNIT-II: 10

Data modeling:Entity relationship (ER) modeling: Entity types, relationships, constraints, ER diagrams, EER model

UNIT-III: 10

Relation data model:Relational model concepts, relational constraints, relational algebra.

UNIT-IV: 15

SQL queries:SQL data definition, data types, specifying constraints, Queries for retrieval, insertion, deletion, updation, introduction to views.

UNIT-V: 15

Database design:Mapping ER/EER model to relational database, functional dependencies,Lossless decomposition, Normal forms (upto BCNF).

UNIT-VI: 15

Transaction and data storage:Introduction to transaction processing: ACID properties, concurrency control; Introduction to indexing structures for files.

CSC410 P: Database Management Systems(Practical)

Create and use the following database schema to answer the given queries.

EMPLOYEE Schema
Field Type NULL KEY
DEFAULT
Eno Char(3) NO PRI NIL
Ename Varchar(50) NO NIL
Job_type Varchar(50) NO NIL
Manager Char(3) Yes FK NIL
Hire_date Date NO NIL
Dno Integer YES FK NIL
Commission Decimal(10,2) YES NIL
Salary Decimal(7,2) NO NIL
DEPARTMENT Schema
Field Type NULL KEY
DEFAULT
Dno Integer No PRI NULL
Dname Varchar(50) Yes NULL
Location Varchar(50) Yes Manipur

Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee withthe Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a commaand name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than$2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in therange of $1500 and $2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 andDept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earncommission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is 'A'.
14. Query to display Name of all employees either have two 'R's or have two 'A's in their nameand are either in Dept No = 30 or their Manger's Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commissionamount is 14 greater than their Salary increased by 5%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date eachemployee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly butwants < 3 * Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case andlength of their name of all the employees whose name starts with 'J', 'A' and 'M'.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an 'A' in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employeesworking at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger's Name and theManager's employee no; along with the Employees' Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. andsalary matches both the department no. And the salary of any employee who earns acommission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (*)signifies $100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the averagesalary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than theaverage salary.

35. Query to display Employee Number and Name for all employees who work in a departmentwith any employee whose name contains a 'T'.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department

References

1. Elmasri, R., & Navathe, S.B. (2015). *Fundamentals of Database Systems.* 7th edition. Pearson Education.

Additional Resources

1. Date, C. J. (2004). *An Introduction to database systems*. 8th edition. Pearson Education.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database System Concepts*. 6th edition. McGrawHill.

Tentative weekly teaching plan is as follows:

Week	Content
1	Introduction to databases: Characteristics of database approach, data models,database system architecture, data independence and data abstraction.
2-3	Entity relationship(ER) modeling: Entity types, relationships, constraints;ERexamples
4-5	Relation data model: Relational model concepts, relational constraints,relational algebra; examples
6-8	SQL queries; examples
9	Database design: Mapping ER/EER model to relational database; examples
10-12	Database design: functional dependencies, Lossless decomposition, Normalforms (upto BCNF); examples
13	Transaction and data storage: Transaction processing:ACID properties,concurrency control; File structure and indexing: Operations on files, File ofUnordered and ordered records
14	File structure and indexing: overview of File organizations, Indexingstructures for files, examples
15	XML databases, noSQLsystems

Semester - V

CSC511: Theory of Computation
Credit: 06
Total Marks: 100 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks)
Workload: 5 Lectures (per week), 1Tutorial(per Week)

Course Objective

This course introduces formal models of computation, namely, finite automaton, pushdownautomaton, and Turing machine; and their relationships with formal languages. Students will alsolearn about the limitations of computing machines.

Course Learning Outcomes

On successful completion of the course, a student will be able to:
1. Design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.
2. Apply pumping lemma to prove that a language is non-regular/non-context-free.
3. Describe limitations of a computing machine.

Detailed Syllabus

UNIT-I: 5

Languages**:** Alphabets, string, language, basic operations on language, concatenation, union, Kleene star.

UNIT-II: 10

Regular Expressions and Finite Automata:Regular expressions, Deterministic finite automata (DFA).

UNIT-III: 15

Regular Languages:Non-deterministic Finite Automata (NFA), relationship between NFA and DFA, Transition Graphs (TG), properties of regular languages, the relationship between regular languages and finite automata, Kleene's Theorem.

UNIT-IV: 10

Non-Regular Languages and Context Free Grammars**:** Pumping lemma for regular grammars, Context-Free Grammars (CFG),

UNIT-V: 20

Context-Free Languages (CFL) and PDA:Deterministic and non-deterministic Pushdown Automata (PDA), parse trees, leftmost derivation, pumping lemma for CFL, properties of CFL. **UNIT-VI: 15**

Turing Machines and Models of Computations:Turing machine as a model of computation, configuration of simple Turing machine, Church Turing Thesis, Universal Turing Machine, decidability, halting problem.

Tutorial:

Tutorials based on theory.

References

1. Cohen, D. I. A. (2011). *Introduction to Computer Theory*. 2nd edition. Wiley India.
2. Lewis, H.R. & Papadimitriou, H. R. (2002). *Elements of the Theory of Computation*. 6th edition. Prentice Hall of India (PHI)

Additional Resources

1. Goodrich, M., Tamassia, R., & Mount, D.M. (2011). *Data Structures and Algorithms Analysis in C++*. 2nd edition. Wiley.
2. Gopalkrishnan, G.L. (2019) *Automata and Computability: A programmer's perspective*. CRC Press.
3. Linz, P. (2016). *An Introduction to Formal Languages and Automata*.6th edition. Jones and Bartlett Learning.

Tentative weekly teaching plan is as follows:

Week	Topics to be covered
1	Languages: Alphabets, string, language, Basic operations on a Language,concatenation, Kleene Star, Kleene closure.
2	Regular Expression: Definition and use of regular expressions, languagesdefined by regular expressions, understanding a regular expression, buildingregular expressions
3	Introduction to finite automata and its relationship with regular expressions,Finite Automata and their languages, deterministic finite automata (DFA).
4	Transition Graphs Relaxing Restrictions on Inputs in TG (Transition Graph),TG vs. FA, Generalized Transition Graphs (GTG), Introduction to Nondeterminism.
5	Kleene's Theorm: Turning TGs and FA to regular expressions and viceversa, Depicting union of two Regular Languages (RL) using an FA,Depicting concatenation (Product) of two RL using an FA.
6	Keene Star of a RL (Regular Language) using an FA, Non-deterministicfinite automata (NFA), relationship between NFA and DFA, converting NFAto DFA.
7	Regular Languages: Complement and intersection of a regular languages,relationship between regular languages and finite automata.
8	Pumping lemma for regular languages.Introduction to context-free languages.
9	Context Free Grammar: Context free grammars, Parse trees.Introduction to Pushdown Automata (PDA).Pushdown Automata: A new Format for FAs, Introduction to PushdownAutomata (PDA).
10	Pushdown Automata: Adding a pushdown stack, design and analysis ofDeterministic PDA, design and analysis of non-deterministic pushdownautomata.
11	Non-Context-Free Languages: Pumping Lemma for Context-FreeLanguages (CFLs), properties of context free languages.
12	Simple Turing machine as a model of computation and its configuration,computing with Turing machine and its working.
13	Building simple Turing machines, combining Turing machines, ChurchTuring Thesis.
14-15	Universal Turing machine, semi-decidability and decidability, recursivelyenumerable and recursive languages, halting problem.

CSC512: Probability Theory and Statistics
Credit: 06
Total Marks: 100 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks)
Workload: 5 Lectures (per week), 1 Tutorial (per week per student)

Course Objectives:

To make the students familiar with the basic statistical concepts and toolswhich are needed to study situations involving uncertainty or randomness. The course intendsto render the students to several examples and exercises that blend their everyday experienceswith their scientific interests.

Course Learning Outcomes: This course will enable the students to learn:

1. Distributions to study the joint behaviour of two random variables.
2. To establish a formulation helping to predict one variable in terms of the other, i.e., correlation and linear regression.
3. Central limit theorem, which helps to understand the remarkable fact that**:** the empiricalfrequencies of so many natural populations, exhibit a bell shaped curve.

Detailed Syllabus

UNIT-I: 20

Probability Functions and Moment Generating Function: Sample space, Probability set function, Real random variables - Discrete and continuous,Cumulative distribution function, Probability mass/density functions, Transformations,Mathematical expectation, Moments, Moment generating function, Characteristic function.

UNIT-II: 20

Univariate Discrete and Continuous DistributionsDiscrete distributions**:** Uniform, Bernoulli, Binomial, Negative binomial, Geometric andPoisson; Continuous distributions**:** Uniform, Gamma, Exponential, Chi-square, Beta andnormal; Normal approximation to the binomial distribution.

UNIT-III: 15

Bivariate Distribution:Joint cumulative distribution function and its properties, Joint probability density function,Marginal distributions, Expectation of function of two random variables, Joint moment generating function, Conditional distributions and expectations.

UNIT-IV: 20

Correlation, Regression and Central Limit Theorem: The Correlation coefficient, Covariance, Calculation of covariance from joint momentgenerating function, Independent random variables, Linear regression for two variables, Themethod of least squares, Bivariate normal distribution, Chebyshev's theorem, Strong law oflarge numbers, Central limit theorem and weak law of large numbers.

References:

1. Hogg, Robert V., McKean, Joseph W., & Craig, Allen T. (2013). *Introduction to Mathematical Statistics* (7th ed.). Pearson Education, Inc.
2. Miller, Irwin & Miller, Marylees. (2014). John E. Freund's *Mathematical Statistics with Applications* (8th ed.). Pearson. Dorling Kindersley (India).
3. Ross, Sheldon M. (2014). *Introduction to Probability Models* (11th ed.). Elsevier Inc. AP.

Semester - VI

CSC613: Artificial Intelligence
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces the basic concepts and techniques of Artificial Intelligence (AI). Thecourse aims to introduce intelligent agents and reasoning, heuristic search techniques, gameplaying, knowledge representation, reasoning with uncertain knowledge.

Course Learning Outcomes

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

1. Identify problems that are amenable to solution by specific AI methods
2. Represent knowledge in Prolog and write code for drawing inferences.
3. Identify appropriate AI technique for the problem at hand
4. Compare strengths and weaknesses of different artificial Intelligence techniques.
5. Sensitive towards development of responsible Artificial Intelligence

Detailed Syllabus

UNIT-I: 7

Introduction:Introduction to artificial intelligence, background and applications, Turing test, rational agents, intelligent agents, structure, behaviour and environment of intelligent agents. **UNIT-II: 8**

Knowledge Representation:Propositional logic, first order predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, scripts, production rules, conceptual graphs.

UNIT-III: 15

Reasoning with Uncertain Knowledge**:** Uncertainty, non-monotonic reasoning, truth maintenance systems, default reasoning and closed world assumption, Introduction to probabilistic reasoning, Bayesian probabilistic inference, introduction to fuzzy sets and fuzzy logic, reasoning using fuzzy logic.

UNIT-IV: 15

Problem Solving and Searching Techniques:Problem characteristics, production systems, control strategies, breadth first search, depth first search, hill climbing and its variations, heuristics search techniques: best first search, A* algorithm, constraint satisfaction problem, means-end analysis.

UNIT-V: 10

Game Playing**:** introduction to game playing, min-max and alpha-beta pruning algorithms. Prolog Programming:Introduction to Programming in Logic (PROLOG), Lists, Operators, basic Input and Output.

UNIT-VI: 10

Understanding Natural Languages**:** Overview of linguistics, Chomsky hierarchy of grammars, parsing techniques.

UNIT-VII: 10

Ethics in AI, Fairness in AI, Legal perspective

CSC613 P: Artificial Intelligence(Practical)

1. Write a prolog program to calculate the sum of two numbers.
2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.
3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.
4. Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series.
5. Write a Prolog program to implement GCD of two numbers.
6. Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.
7. Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result.
8. Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.
9. Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.
10. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.
11. Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.
12. Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement nth_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L.
15. Write a Prolog program to implement maxlist(L, M) so that M is the maximum number in the list.
16. Write a prolog program to implement insert_nth (I, N, L, R) that inserts an item I into Nth position of list L to generate a list R.
17. Write a Prolog program to implement delete_nth (N, L, R) that removes the element on Nth position from a list L to generate a list R.
18. Write a program in PROLOG to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.

References

1. Rich, E. & Knight,K. (2012). *Artificial Intelligence*. 3rd edition. Tata McGraw Hill.
2. Russell, S.J. & Norvig, P. (2015) *Artificial Intelligence - A Modern Approach*. 3rd edition. Pearson Education
61

Additional Resources:

1. Bratko, I. (2011). *Prolog Programming for Artificial Intelligence*. 4th edition. Pearson Education
2. Clocksin, W.F. & Mellish (2003), *Programming in PROLOG*. 5th edition. Springer
3. Kaushik, S. (2011). *Artificial Intelligence*. Cengage Learning India.
4. Patterson, D.W. (2015). *Introduction to Artificial Intelligence and Expert Systems.* 1st edition. Pearson Education.

Tentative weekly teaching plan is as follows:

Week	Content
1	Introduction to artificial intelligence, background and applications, Turing test and rational agent approaches to AI, introduction to intelligent agents.
2	Structure, behavior and environment of intelligent agents, problem characteristics, production systems, control strategies.
3	Introduction to programming in logic (PROLOG).
4	Programming in logic (PROLOG), breadth first search, depth firstsearch introduction of heuristic search techniques.
5	Propositional logic, first order predicate logic.
6	Unification, clausal form, resolution principle.
7	Semantic nets, conceptual graphs, conceptual dependencies.
8	Frames, scripts, Uncertainty: non-monotonic reasoning, truthmaintenance systems, default reasoning and closed worldassumption.
9	Bayesian probabilistic inference, Bayesian networks, DempsterShafer theory, Introduction to fuzzy sets and fuzzy logic.
10	Basic reasoning using fuzzy concepts, production rules, Chomskyhierarchy of grammars, context-free grammars.
11	Hill climbing and its variations, best first search.
12	A* algorithm, constraint satisfaction problem, means-end analysis.
13	Introduction to game playing, min-max procedure, alpha-betapruning.
14-15	Overview of linguistics, Chomsky hierarchy of grammars, parsingtechniques..

CSC614: Computer Graphics
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces fundamental concepts of Computer Graphics with focus on modelling,rendering and interaction aspects of computer graphics. The course emphasizes the basicprinciples needed to design, use and understand computer graphics system.

Course Learning Outcomes

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

1. Describe Standard raster and vector scan devices as well as Graphical Input and output devices
2. Implement algorithms for drawing basic primitives such as linecircle and ellipse.
3. Implement algorithms for line clipping and polygon clipping and filling.
4. Implement a 3D object representation scheme and carryout 2D and 3D Transformation, 3D projections
5. Implement visible surface determination algorithms, Illumination models and surface rendering methods, color models
6. Implement a simple computer animation algorithm

Detailed Syllabus

UNIT-I: 10

Introduction:Introduction to Graphics systems, Basic elements of Computer graphics, Applications of computer graphics. Architecture of Raster and Random scan display devices, input/output devices.

UNIT-II: 20

Drawing and clipping primitives:Raster scan line, circle and ellipse drawing algorithms, Polygon filling, line clipping and polygon clipping algorithms

UNIT-III: 15

Transformation and Viewing:2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.

UNIT-IV: 15

Geometric Modeling:Polygon Mesh Representation, Cubic Polynomial curves (Hermite and Bezier).

UNIT-V: 15

Visible Surface determination and Surface Rendering:Z-buffer algorithm, List-priority algorithm and area subdivision algorithm for visible surface determination. Illumination and shading models, RGB color model and Basics of Computer Animation.

CSC614 P: Computer Graphics(Practical)

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement mid-point circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to fill a polygon using Scan line fill algorithm.
6. Write a program to apply various 2D transformations on a 2D object (use homogenous Coordinates).
7. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.
8. Write a program to draw Hermite /Bezier curve.

References

1. Baker, D.H. (2008). *Computer Graphics.* 2nd edition. Prentice Hall of India.
2. Foley, J. D., Dam, A.V, Feiner, S. K., & Hughes, J. F. (1995). *Computer Graphics: Principles and Practice in C.* 2nd edition. Addison-Wesley Professional.

Additional Resources:

1. Bhattacharya, S. (2018). *Computer Graphics*. Oxford University Press
2. Cohen, D. I. A. (2011). *Introduction to Computer Theory*. 2nd edition. Wiley India.
3. Marschner, S., & Shirley, P. (2017) *Fundamentals of Computer Graphics*. 4th edition. CRC Press
4. Rogers, D. F. (1989). *Mathematical Elements for Computer Graphics*. 2nd edition. McGraw Hill.

Tentative weekly teaching plan is as follows:

Week	Contents
1	Introduction to Graphics systems, Basic elements of Computer graphics, Applications of computer graphics.
2	Graphics Hardware: Architecture of Raster and Random scan display devices, input/output devices.
3-4	Drawing Primitives: Raster scan line drawing algorithm, circle and ellipse drawing algorithms
5	Polygon filling, line clipping and polygon clipping algorithms
6	Transformation: 2D and 3D Geometric Transformations
7-9	Viewing : 3D Viewing Transformations, Parallel Projections, Perspective Projections ,Vanishing points
10	Geometric Modeling: Representing curves(Hermite and Bezier)
11-12	Geometric Modeling: Representing curves(Hermite and Bezier)(cont.), Visible Surface determination: Z-buffer algorithm
13	List-priority algorithm and area subdivision algorithm.
14	Surface rendering: Illumination and shading models
15	RGB color model and Computer Animation.

Semester - VII

CSC715:Software Engineering
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

The course introduces fundamental Software Engineering approaches and techniques forsoftware development. The students also develop a case study using appropriate software model.

Course Learning Outcomes

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

1. Analyse and model customer's requirements and model its software design.
2. Use suitable software model for the problem at hand.
3. Estimate cost and efforts required in building software.
4. Analyse and compute impact of various risks involved in software development.
5. Design and build test cases, and to perform software testing.

Detailed Syllabus

UNIT-I: **10**

Introduction:Software Engineering - A Layered Approach; Software Process – ProcessFramework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, andEvolutionary process Model (Prototyping, Spiral Model); Introduction to Agile – Agility Principles, Agile Model – Scrum.

UNIT-II: **10**

Software Requirements Analysis and Specifications:Use Case Approach, Software Requirement Specification Document, Flow oriented Modeling, Data Flow Modeling, Sequence Diagrams

UNIT-III: **10**

Design Modeling:Translating the Requirements model into the Design Model, The Design Process, Design Concepts - Abstraction, Modularity and Functional Independence; Architectural Mapping using Data Flow.

UNIT-IV: **15**

Software Metrics and Project Estimations:Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations, COCOMO II Model); Project Scheduling (Timeline charts, tracking the schedule).

UNIT-V: **15**

Quality Control and Risk Management:Quality Control and Quality Assurance, Software Process Assessment and Improvement Capability Maturity Model Integration (CMMI); Software Risks, Risk Identification, Risk Projection and Risk Refinement, Risk Mitigation, Monitoring and Management.

UNIT-VI: **15**

Software Testing:Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing.

CSC715 P: Software Engineering(Practical)

Practical problems related to

1. Requirement Analysis, Creating a Data Flow, Data Dictionary, Use Cases
2. Computing FP, Effort, Schedule, Risk Table, Timeline chart
3. Design Engineering, Architectural Design, Data Design, Component Level Design
4. Testing, Basis Path Testing

Sample Projects:
1. Criminal Record Management: Implement a criminal record management system for jailers, police officers and CBI officers
2. DTC Route Information: Online information about the bus routes and their frequency and fares
3. Car Pooling: To maintain a web based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

References

1. Aggarwal, K. K., & Singh, Y. (2007). *Software Engineering*. 3rd edition. New Age International Publishers.
2. Pressman, R. S., & Maxim, B. R. (2015). *Software Engineering: A Practitioner's Approach*. 8th edition. McGraw-Hill.

Additional Resources

1. Jalote, P. (2005). *An Integrated Approach to Software Engineering*. 3rd edition. Narosa Publishing House.
2. Schwaber, K. & Sutherland, J. (2016). *The Definitive Guide to Scrum: The Rules of the Game*. [https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf]
3. Sommerville. (2011). *Software Engineering*. 9th edition. Addison Wesley.

Tentative weekly teaching plan is as follows:

Week	Content
1	Software - Nature of Software, Software Application Domains, LegacySoftware; Software Engineering - A Layered Approach; Software Process –Process Framework, Framework and Umbrella Activities
2	Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model);
3	Introduction to Agile – Agility, Cost of Change, Agility Principles
4	Agile Model - Scrum; Software Process Assessment and Improvement - Capability Maturity Model Integration (CMMI).
5	Requirements Modelling - Requirements Modelling Approaches, Flow orientedModelling, Data Flow Modelling,

6	Control Flow Model, Control Specification, Process Specification, Behavioural Model, State Diagram, Sequence Diagrams;
7	Design Modelling - Design Concepts, Translating requirements model intodesign model, Design Process, Abstraction, Architecture, Separation ofconcerns, Modularity, Information hiding, Functional Independence,
8	Refinement, Refactoring; Architectural Mapping using Data Flow.
9	Risk Management- Software Risks, Risk Identification, Risk Projection and Risk Refinement, Risk Mitigation, Monitoring and Management.
10	Function based Product Metrics, Software Quality Metrics;
11	Estimation for Software Project, Project Scheduling, Quality – SoftwareQuality, McCall's Quality Factors, ISO 9126 Quality Factors, AchievingSoftware Quality;
12	Cost Impact of Software Defects, Defect Amplification and Removal, FormalTechnical Reviews; Software Quality Assurance – SQA Tasks.
13-14	Software Testing - Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing;
15	Black-Box and White Box Testing, Basis Path Testing

CSC716: Data Mining
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces data mining techniques and enables students to apply these techniques onreal-life datasets. The course focuses on three main data mining techniques: Classification,Clustering and Association Rule Mining tasks.

Course Learning Outcomes

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

1. Pre-process the data, and perform cleaning and transformation.
2. Apply suitable classification algorithm to train the classifier and evaluate its performance.
3. Apply appropriate clustering algorithm to cluster data and evaluate clustering quality
4. Use association rule mining algorithms and generate frequent item-sets and association rules

Detailed Syllabus

UNIT-I: 20

Introduction to Data Mining- Applications of data mining, data mining tasks, motivation and challenges, types of data attributes and measurements, data quality.

Data Pre-processing - aggregation, sampling, dimensionality reduction, Feature Subset Selection, Feature Creation, Discretization and Binarization, Variable Transformation.

UNIT-II: 20

Classification:Basic Concepts, Decision Tree Classifier: Decision tree algorithm, attribute selection measures, Nearest Neighbour Classifier, Bayes Theorem and Naive Bayes Classifier,

Model Evaluation:Holdout Method, Random Sub Sampling, Cross-Validation, evaluation metrics, confusion matrix.

UNIT-III: **15**

Association rule mining: Transaction data-set, Frequent Itemset, Support measure, Apriori Principle, Apriori Algorithm, Computational Complexity, Rule Generation, Confidence of association rule.

UNIT-IV: **20**

Cluster Analysis:Basic Concepts, Different Types of Clustering Methods, Different Types of Clusters, K-means: The Basic K-means Algorithm, Strengths and Weaknesses of K-means algorithm, Agglomerative Hierarchical Clustering: Basic Algorithm, Proximity between clusters, DBSCAN: The DBSCAN Algorithm, Strengths and Weaknesses.

CSC716 P: Data Mining(Practical)

Section 1: Pre-processing

Q1. Create a file "people.txt" with the following data:

Age	agegroup	height	status	yearsmarried
21	adult	6.0	single	-1
2	child	3	married	0
18	adult	5.7	married	20
221	elderly	5	widowed	2
34	child	-7	married	3

i) Read the data from the file "*people.txt".*

ii) Create a ruleset *E* that contain rules to check for the following conditions:

1. The age should be in the range 0-150.
2. The age should be greater than yearsmarried.
3. The status should be married or single or widowed.
4. If age is less than 18 the agegroup should be child, if age is between 18 and 65 the agegroup should be adult, if age is more than 65 the agegroup should be elderly.

iii) Check whether ruleset *E* is violated by the data in the file people.txt.

iv) Summarize the results obtained in part (iii)

v) Visualize the results obtained in part (iii)

Q2. Perform the following pre-processing tasks on the dirty_iris datasetii.

1. Calculate the number and percentage of observations that are complete.
2. Replace all the special values in data with NA.
3. Define these rules in a separate text file and read them.

(Use editfile function in R (package editrules). Use similar function in Python).

Print the resulting constraint object.

– Species should be one of the following values: setosa, versicolor or virginica.
– All measured numerical properties of an iris should be positive.
– The petal length of an iris is at least 2 times its petal width.
– The sepal length of an iris cannot exceed 30 cm.
– The sepals of an iris are longer than its petals.

4. Determine how often each rule is broken (violatedEdits). Also summarize and plot the result.

Find outliers in sepal length using boxplot and boxplot.stats

Q3. Load the data from wine dataset. Check whether all attributes are standardized or not (mean is 0 and standard deviation is 1). If not, standardize the attributes. Do the same with Iris dataset.

Section 2: Data Mining Techniques

Run following algorithms on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns:

Q4. Run Apriori algorithm to find frequent itemsets and association rules

4.1 Use minimum support as 50% and minimum confidence as 75%

4.2 Use minimum support as 60% and minimum confidence as 60 %
Q5. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations:
5.1 a) Training set = 75% Test set = 25%
b) Training set = 66.6% (2/3rd of total), Test set = 33.3%
5.2 Training set is chosen by i) hold out method ii) Random subsampling iii) CrossValidation. Compare the accuracy of the classifiers obtained.
5.3 Data is scaled to standard format.
Q6. Use Simple Kmeans, DBScan, Hierachical clustering algorithms for clustering. Compare the performance of clusters by changing the parameters involved in the algorithms.

Recommended Datasets for DataMining practicals

1. UCI Machine Learning repository.
2. KDD Datasets
3. Open data platform, Government of India (https://data.gov.in/)

References

1. Han, J., Kamber, M.,& Jian,P. (2011). *Data Mining: Concepts and Techniques*. 3rd edition. Morgan Kaufmann
2. Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. 1st Edition. Pearson Education.

Additional Resources

1. Gupta, G. K. (2006). *Introduction to Data Mining with Case Studies*. Prentice-Hall of India.
2. Hand, D., & Mannila, H. & Smyth, P. (2006). *Principles of Data Mining*. Prentice-Hall of India.
3. Pujari, A. (2008). *Data Mining Techniques*. 2nd edition. Universities Press.

Tentative weekly teaching plan is as follows:

Week	Content
1	Introduction to Data Mining , Challenges , Data Mining Origins, Data Mining Tasks, Applications
2-3	Types of data, Data Quality, Data Pre-processing, Measures of similarity anddissimilarity
5-8	Classification - Preliminaries, General Approach to Solving a ClassificationProblem, Decision Tree Induction , Evaluating the Performance of a Classifier
8-9	Rule Based Classifier , Nearest Neighbor Classifiers, Bayesian Classifiers
10-11	Association Rules -Problem definition, Frequent item-set generation (Apriorialgorithm), Rule generation
11-12	Clustering - Basic concepts of clustering analysis, K-Means
13-14	Agglomerative Hierarchical Clustering, DBSCAN
15	Quality of clustering

Semester - VIII

CSC817: Information Security
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

The course offers a broad overview of the fundamentals of information security covering topicssuch as error correction/detection, cryptography, steganography, malwares. This course alsotouches on the implications of security in Internet of Things (IoT).

Course Learning Outcomes

On successful completion of this course, a student will be able to:

1. Identify the major types of threats to information security
2. Describe the role of cryptography in security
3. Select appropriate error-detection and error-correction methods for an application
4. Discuss the strengths and weaknesses of private and public key crypto systems
5. Describe malwares and memory exploits
6. Discuss the need for security in IoT

Detailed Syllabus

UNIT-I: 10

Introduction:Security Concepts, Challenges, Security architecture, Security attacks, security services, security mechanisms

UNIT-II: 15

Error detecting/correction:Block Codes, Generator Matrix, Parity Check Matrix, Minimum distance of a Code, Error detection and correction, Standard Array and syndrome decoding, Hamming Codes

UNIT-III: 20

Cryptography:Encryption, Decryption, Substitution and Transposition, Confusion and diffusion, Symmetric and Asymmetric encryption, Stream and Block ciphers, DES, cryptanalysis.Public-key cryptography, Diffie-Hellman key exchange, man-in-the-middle attack Digital signature, Steganography, Watermarking.

UNIT-IV: 15

Malicious software's: Types of malwares (viruses, worms, trojan horse, rootkits, bots), Memory exploits - Buffer overflow, Integer overflow

UNIT-V: 15

Security in Internet-of-Things:Security implications, Mobile device security - threats and strategies

Practical

1. Implement the error correcting code.
2. Implement the error detecting code.
3. Implement caeser cipher substitution operation.
4. Implement monoalphabetic and polyalphabetic cipher substitution operation.
5. Implement playfair cipher substitution operation.
6. Implement hill cipher substitution operation.
7. Implement rail fence cipher transposition operation.

8. Implement row transposition cipher transposition operation.
9. Implement product cipher transposition operation.
10.Illustrate the Ciphertext only and Known plaintext attacks.
11.Implement a stream cipher technique

References

1. Pfleeger, C.P., Pfleeger,S.L., & Margulies, J. (2015). *Security in Computing.* 5th edition. Prentice Hall
2. Lin, S. & Costello, D. J. (2004). *Error Control Coding: Fundamentals and applications*. 2nd edition. Pearson Education
3. Stallings, W. (2018). *Cryptography and network security*. 7th edition. Pearson Education.

Additional Resources

1. Berlekamp, E. R. (1986). *Algebraic Coding Theory*. McGraw Hill Book Company
2. Stallings, W. (2018) *Network security, essentials.* 6th edition. Pearson Education.
3. Whitman M.E., & Mattord H.J. (2017). *Principle of Information Security*. 6th edition. Cengage Learning.

Tentative weekly teaching plan is as follows:

Week	Content
1-2	Security Concepts, Challenges, Security architecture, Securityattacks, security services, security mechanisms
3-4	Error detecting/correction, Block Codes, Generator Matrix, ParityCheck Matrix, Minimum distance of a Code, Error detection andcorrection, Standard Array and syndrome decoding, HammingCodes
5-7	Cryptography: Encryption, Decryption, Substitution andTransposition, Confusion and diffusion, Symmetric andAsymmetric encryption, Stream and Block ciphers, DES, Modes ofDES
8-9	Cryptanalysis, Types of cryptanalytic attacks, Public-keycryptography, Diffie-Hellman key exchange, man-in-the-middleattack
10-11	Digital signatures, Steganography and Digital Watermarking
12-13	Malicious Software: Types of malwares (viruses, worms, Trojan horse, rootkits, bots), Memory exploits - Buffer overflow, Integeroverflow
14-15	Security in Internet-of-Things, Security implications, Mobile devicesecurity - threats and strategies, Cyberlaws

CSC818: Digital Image Processing
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces students to the fundamentals of digital image processing, and variousimage transforms, image restoration techniques, image compression and segmentation used indigital image processing.

Course Learning Outcomes

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

1. Describe the roles of image processing systems in a variety of applications.
2. Write programs to read/write and manipulate images: enhancement, segmentation, and compression, spatial filtering.
3. Develop Fourier transform for image processing in frequency domain.
4. Evaluate the methodologies for image segmentation, restoration

Detailed Syllabus

UNIT-I: 15

Introduction:Digital Image Fundamentals: Brightness, Adaptation and Discrimination, Light and Electromagnetic Spectrum, Image Sampling and Quantization, Some Basic Relationships between Pixels Types of images.

UNIT-II: 15

Spatial Domain Filtering:Some Basic Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution, Smoothening Spatial Filters: Low pass filters, Order Statistics filters; Sharpening Spatial Filters: Laplacian filter **UNIT-III: 10**

Filtering in Frequency Domain:The Discrete Fourier Transformation (DFT), Frequency Domain Filtering: Ideal and Butterworth Low pass and High pass filters, DCT Transform (1D, 2D).

UNIT-IV: 10

Image Restoration:Image Degradation/Restoration Process, Noise models, Noise Restoration Filters. Image Compression:Fundamentals of Image Compression, Huffman Coding, Run Length Coding, JPEG.

UNIT-V: 15

Morphological Image Processing:Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Basic Morphological Algorithms.

UNIT-VI: 10

Image Segmentation:Point, Line and Edge Detection, Thresholding, Region BasedSegmentation.

CSC818 P: Digital Image Processing(Practical)

1. Write program to read and display digital image using MATLAB or SCILAB
a. Become familiar with SCILAB/MATLAB Basic commands
b. Read and display image in SCILAB/MATLAB
c. Resize given image
d. Convert given color image into gray-scale image
e. Convert given color/gray-scale image into black & white image
f. Draw image profile
g. Separate color image in three R G & B planes
h. Create color image using R, G and B three separate planes
i. Flow control and LOOP in SCILAB
j. Write given 2-D data in image file

2. To write and execute image processing programs using point processing method
a. Obtain Negative image
b. Obtain Flip image
c. Thresholding
d. Contrast stretching

3. To write and execute programs for image arithmetic operations
a. Addition of two images
b. Subtract one image from other image
c. Calculate mean value of image
d. Different Brightness by changing mean value

4. To write and execute programs for image logical operations
a. AND operation between two images
b. OR operation between two images
c. Calculate intersection of two images
d. Water Marking using EX-OR operation
e. NOT operation (Negative image)

5. To write a program for histogram calculation and equalization using
a. Standard MATLAB function
b. Program without using standard MATLAB functions
c. C Program

6. To write and execute program for geometric transformation of image
a. Translation
b. Scaling
c. Rotation
d. Shrinking
e. Zooming

7. To understand various image noise models and to write programs for
a. image restoration
b. Remove Salt and Pepper Noise
c. Minimize Gaussian noise
d. Median filter and Weiner filter

8. Write and execute programs to remove noise using spatial filters
a. Understand 1-D and 2-D convolution process
b. Use 3x3 Mask for low pass filter and high pass filter

9. Write and execute programs for image frequency domain filtering
a. Apply FFT on given image
b. Perform low pass and high pass filtering in frequency domain
c. Apply IFFT to reconstruct image

10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask
11. Write and execute program for image morphological operations erosion and dilation.
12. To write and execute program for wavelet transform on given image and perform inverse wavelet transform to reconstruct image.

References

1. Gonzalez, R. C., & Woods, R. E. (2017). *Digital Image Processing*. 4th edition. Pearson Education.
2. Jain, A. K. (1988). *Fundamentals of Digital Image Processing*. 1st edition Prentice Hall of India.

Additional Resources

1. Castleman, K. R. (1995.). *Digital Image Processing*. 1st edition. Pearson Education
2. Gonzalez, R. C., Woods, R. E., & Eddins, S. (2004). *Digital Image Processing using MATLAB*. Pearson Education Inc.
3. Schalkoff, D. (1989). *Image Processing and Computer Vision*. 1st edition. John Wiley and Sons.

Tentative weekly teaching plan is as follows:

Week	**Content**
1	Brightness, Adaptation and Discrimination, Light and Electromagnetic Spectrum, Image Sampling and Quantization.
2-5	Some Basic Relationships Between Pixels ,Spatial Domain Filtering, Intensity Transformation Functions, Histogram Equalization, Spatial Correlation and Convolution , Low pass filters, Order Statistics filters, Sharpening Spatial Filters: Laplacian filterFiltering in Frequency Domain The Discrete Fourier Transformation(DFT)
6-7	Frequency Domain Filtering:Ideal and Butterworth Low pass and High pass filters, Image Degradation/Restoration Process
8-10	Noise models, Noise Restoration Filters, Image Compression, Huffman Coding,Run Length Coding, Bit Plane Coding
11-12	Morphological Image Processing, Erosion, Dilation, Opening, Closing , Hit-or-Miss Transformation, Basic Morphological Algorithms
13-15	Image Segmentation: Point, Line and Edge Detection ,Thresholding, Region Based Segmentation

Semester- V
Discipline Specific Elective Papers

CSE501A: Data Analysis and Visualization
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces students to data analysis and visualization in the field of exploratory datascience using Python.

Course Learning Outcomes

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

1. Use data analysis tools in the pandas library.
2. Load, clean, transform, merge and reshape data.
3. Create informative visualization and summarize data sets.
4. Analyze and manipulate time series data.
5. Solve real world data analysis problems.

Detailed Syllabus

UNIT-I: **10**

Introduction:Introduction to Data Science, Exploratory Data Analysis and Data ScienceProcess. Motivation for using Python for Data Analysis, Introduction of Python shell iPythonand Jupyter Notebook.

Essential Python Libraries:NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels

UNIT-II: **20**

Getting Started with Pandas: Arrays and vectorized conputation, Introduction to pandas Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics. Data Loading, Storage and File Formats.Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, Interacting withWeb APIs, Interacting with DatabasesData Cleaning and Preparation.Handling Missing Data, Data Transformation, String Manipulation.

UNIT-III: **15**

Data Wrangling:Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting.

Data Visualization matplotlib: Basics of matplotlib, plotting with pandas and seaborn, other python visualization tools

UNIT-IV: **20**

Data Aggregation and Group operations: Group by Mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation

Time Series Data Analysis: Date and Time Data Types and Tools, Time series Basics, date Ranges, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency conversion, Moving Window Functions.

UNIT-V: **10**

Advanced Pandas: Categorical Data, Advanced GroupBy Use, Techniques for Method Chaining

CSE501A: Data Analysis and Visualization(Practical)

Use data set of your choice from Open Data Portal (https://data.gov.in/) for the following exercises.

1. Practicals based on NumPy ndarray
2. Practicals based on Pandas Data Structures
3. Practicals based on Data Loading, Storage and File Formats
4. Practicals based on Interacting with Web APIs
5. Practicals based on Data Cleaning and Preparation
6. Practicals based on Data Wrangling
7. Practicals based on Data Visualization using matplotlib
8. Practicals based on Data Aggregation
9. Practicals based on Time Series Data Analysis

References

1. McKinney, W.(2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython*. 2nd edition. O'Reilly Media.

2. O'Neil, C., & Schutt, R. (2013). *Doing Data Science: Straight Talk from the Frontline* O'Reilly Media

Tentative weekly teaching plan is as follows:

Week	Content
1	Introduction: What is Data Science? Exploratory Data Analysis and Data Science Process. Why Python for Data Analysis? Introduction of Python shell iPython andJupyter Notebook.
2-3	Essential Python Libraries: Learn NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels.
4	Built-in Data Structures, Function and Files: Data Structure and sequences, Functions, Files and Operating systems
5	Arrays and Vectorized computation: The NumPy ndarray, Universal Functions,Array Oriented Programming with Arrays, File Input and Output with Arrays,Linear Algebra, Pseudorandom Number Generation
6	Getting Started with pandas: Introduction to pandas Data Structures, EssentialFunctionality, Summarizing and Computing Descriptive Statistics.
7	Data Loading, Storage and File Formats: Reading and Writing Data in Text Format, Web Scraping, Binary Data Formats, Interacting with Web APIs, Interacting with Databases.
8	Data Cleaning and Preparation: Handling Missing Data, Data Transformation,String Manipulation
9	Data Wrangling: Hierarchical Indexing, Combining and Merging Data SetsReshaping and Pivoting.
10	Data Visualization matplotlib: Basics of matplotlib, plotting with pandas andseaborn, other python visualization tools.
11	Data Aggregation and Group operations: Group by Mechanics, Data aggregation,General split-apply-combine, Pivot tables and cross tabulation
12-13	Time Series Data Analysis: Date and Time Data Types and Tools, Time seriesBasics, date Ranges, Frequencies and Shifting, Time Zone Handling, Periods andPeriods Arithmetic, Resampling and Frequency conversion, Moving WindowFunctions
14-15	Data Analysis Case Studies

CSE501B: Operational Research for Computer Science
Credit: 06
Total Marks: 100 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks)
Workload: 5 Lectures (per week), 1 Tutorial(per Week)

UNIT-I: 5

Introductory Linear Algebra:System of linear equations, Matrices, Rank and Determinant of a matrix, Linearly dependent and independent vectors, Basis of a matrix.

UNIT-II: 15

Linear programming – I:Optimization Problems, Introduction to LP Formulation, Convex sets, Extreme points, Geometry ofLinear Programs, Basic feasible solutions (BFS), Neighborhoods, Local and global optima, ProfitableColumn, Pivoting, Simplex Algorithm with initial BFS, Graphical method.

UNIT-III: 15

Linear programming – II:Degeneracy and Bland's Anticycling rule (Definition), Simplex Algorithm without initial BFS,Artificial variable techniques – two phase method, M-Charnes method, special cases in LPP.

UNIT-IV: 10

Duality: Definition of the dual problem, primal-dual relationships, economic interpretation of duality,complementary slackness conditions.

UNIT- V: 10

Transportation Models:Transportation Algorithm, Assignment model, Hungarian Method

UNIT-VI: 10

Introduction to Queuing Models:Elements of Queuing Model, Exponential distribution, Poisson Distributions, Poisson QueuingModels, Single Server model, Multiple Server model

UNIT-VII: 10

Introduction to Markov Chains:Introduction to Markov chains, transition probabilities, classification of states, Steady stateprobabilities, Absorbing states

Tutorial

Tutorial based on theory

Reference Books

1. G. Hadley: Linear Programming. Narosa, 2002 (reprint).
2. A. Ravindran, D. T. Phillips and James J. Solberg: Operations Research-Principles and Practice, John Wiley & Sons, 2005.
3. Hamdy A. Taha: Operations Research-An Introduction, Prentice Hall, 8th Edition, 2008.
4. F.S. Hillier. G.J. Lieberman: Introduction to Operations Research- Concepts and Cases, 9th Edition, Tata McGraw Hill. 2010.

CSE501C: Computer Oriented Numerical Methods
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

UNIT-I: 15

Errors & Their Accuracy, Solutions of Algebraic and Transcendental Equations, Bisection Method,
The method of false position, The iteration method, Newton Raphson method, Generalized Newton's method, Solutions of system of non-linear Equations: The method of iteration, Newton Raphson method.

UNIT-II: 20

Interpolation Errors in polynomial interpolation, Finite Differences Detection of errors by use of difference tables, Differences of a Polynomial, Newton's formulae for interpolation, Central difference interpolation formulae, Guass's Central difference formulae, Interpolation with unevenly spaced points Lagrange's interpolation, Error in Lagrange's interpolation, Divided differences and their properties, Newton's general interpolation formula-interpolation by iteration, inverse interpolation.

UNIT-III: 20

Differentiation and Integration, Numerical Differentiation, Methods based on interpolation, Nonuniform nodal points-Linear interpolation, quadratic interpolation Uniform nodal points-Linear interpolation, quadratic interpolation Methods based on finite differences Numerical Integration Methods based on interpolation-Newton's Cotes methods, Trapezoidal method, Simpson's method,3/8 Simpson's rule, open type integration rules. Methods based on undetermined coefficients - Newton's methods, Trapezoidal rule, Simpson's rule.

UNIT-IV: 20

Solution of System of linear equation by iteration method, Gauss-Sidel method, Jacobi's method, Numerical solution of Ordinary Differential Equation. Solution by Taylor's series, Euler's method, Modified Euler's method, Runge-Kutta method.

Books:

1. Introductory methods of Numerical Analysis - By S. S. Sastry, PHI
2. Numerical Methods for Scientific and Engineering students - M.K.Jain, S.R.K.Iyengar, New age international (P) Ltd.
3. Computer Oriented Numerical Methods, - V. Rajaraman, PHI

CSE501C P: Computer Oriented Numerical Methods(Practical)

1. To find out the root of the Algebraic and Transcendental equations using Bisection method.
2. To find out the root of the Algebraic and Transcendental equations using Newton-Raphson method.
3. To implement Numerical Integration using Trapezoidal rule.
4. To implement Numerical Integration using Simpson 1/3 rule.
5. To implement Numerical Integration Simpson 3/8 rule.

Semester- VI
Discipline Specific Elective Papers

CSE601A: System Programming
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

The course is focused on design of assembler and basic compiler. The course covers topics likeabsolute loader, relocating loader and dynamic linking.

Course Learning Outcomes

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

1. Describe the working of assemblers and compilers.
2. Use Lex/ Yacc for building basic compiler.
3. Develop a two pass Assemblers.
4. Describe the role of the loaders, linkers and relocatable programs

.

Detailed Syllabus

UNIT-I: **10**

Assemblers & Loaders, Linkers: One pass and two pass assembler, design of an assembler,Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking.

UNIT-II: **10**

Introduction:Overview of compilation, Phases of a compiler.

UNIT-III: **20**

Lexical Analysis:Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lexical Analyzer Generator.

UNIT-IV: **20**

Parsing & Intermediate representations:Bottom up parsing- LR parser, yacc,three address code generation, syntax directed translation, translation of types, control statements

UNIT-V: **15**

Storage organization & Code generation**:** Activation records, stack allocation, Object code generation

CSE602A P: System Programming(Practical)

Projects to implement an assembler for a hypothetical language.

Programs to get familiar with Lex and Yacc

1. Write a Lex program to count the number of lines and characters in the input file.
2. Write a Lex program that implements the Caesar cipher: it replaces every letter with the one three letters after in in alphabetical order, wrapping around at Z. e.g. a is replaced by d, b by e, and so on z by c.
3. Write a Lex program that finds the longest word (defined as a contiguous string of upper and lower case letters) in the input.

4. Write a Lex program that distinguishes keywords, integers, floats, identifiers, operators, and comments in any simple programming language.

5. Write a Lex program to count the number of identifiers in a C file.
6. Write a Lex program to count the number of words, characters, blank spaces and lines in a C file.
7. Write a Lex specification program that generates a C program which takes a string "abcd" and prints the following output
abcd
abc
a
8. A program in Lex to recognize a valid arithmetic expression.
9. Write a YACC program to find the validity of a given expression (for operators + - * and /)A program in YACC which recognizes a valid variable which starts with letter followed by a digit. The letter should be in lowercase only.
10. A Program in YACC to evaluate an expression (simple calculator program for addition and subtraction, multiplication, division).
11. Program in YACC to recognize the string „abbb", „ab" „a" of the langauge (an b n , n>=1).
12. Program in YACC to recognize the language (an b , n>=10). (output to say input is valid or not)

References

1. Aho, A., Lam, M., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools*. 2nd edition. Addison Wesley.
2. Chattopadhyaya, S. (2011). *System Software*. P H I Learning.

Additional references:

1. Beck, L. & Manjula, D. (1996). *System Software: An Introduction to System Programming*. 3rd edition. Pearson Education.
2. Dhamdhere, D. M. (2015). *Systems Programming*. Tata McGrawHill.

Tentative weekly teaching plan is as follows:

Week	Content
1-3	Assemblers & Loaders, Linkers:One pass and two pass assembler, design of an assembler, Absolute loader, relocation and linking concepts, relocatingloader and Dynamic Linking.
4	Overview of compilation, Phases of acompiler.
5-6	Lexical Analysis: Role of a Lexical analyzer, Specificationand recognition of tokens,Symbol table, lexical AnalyzerGenerator.
7-9	Parsing : Bottom up parsing- LR parser,yacc.
10-11	Intermediate representations: Three address codegeneration,syntax directed translation, translation of types,control statements
12-15	Storage organization & Code generation: Activationrecords, stack allocation, Object code generation

CSE602B:Microprocessors
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces internal architecture, programming model of Intel Microprocessors (8086

-Pentium) and assembly language programming using an assembler. Students will also learn interfacing of memory and I/O devices with microprocessor.

Course Learning Outcomes

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

1. Describe the internal architecture of Intel microprocessors
2. Define and implement interfaces between the microprocessor and the devices.
3. Write assembly language programs

Detailed Syllabus

UNIT-I: 10

Microprocessor architecture**:** Internal architecture, Programming Model, Addressing modes, Data movement instructions

UNIT-II: 10

Microprocessor programming:Register Organization, instruction formats, Program control instructions, assembly language

UNIT-III: 15

Interfacing**:** Bus timings, Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, timer, Interrupt controller, DMA controller, video controllers, communication interfaces.

UNIT-IV: 15

Data transfer schemes:Synchronous data transfer, asynchronous data transfer, interrupt driven data transfer, DMA mode data transfer.

UNIT-V: 10

Microprocessor controllers:I/O controllers, interrupt controller, DMA controller, USART controller.

UNIT-VI: 15

Advance microprocessor architecture:CISC architecture, RISC architecture, superscalar architecture, multicore architecture

CSE602B P:Microprocessors(Practical)

ASSEMBLY LANGUAGE PROGRAMMING

1. Write a program for 32-bit binary division and multiplication
2. Write a program for 32-bit BCD addition and subtraction
3. Write a program for Linear search and binary search.
4. Write a program to add and subtract two arrays
5. Write a program for binary to ascii conversion
6. Write a program for ascii to binary conversion

References

1. Brey, B.B.(2009). *The Intel Microprocessors: Architecture, Programming and Interfacing*. 8th edition. Pearson Education.

2. Triebel, W.A., & Singh, A. (2002). *The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware and Applications.* 4th edition. Pearson Education

Tentative weekly teaching plan is as follows:

Week	Content
1-2	Microprocessor Architecture: Internal Architecture of microprocessor, RegisterOrganization and flags, Programming models, Real mode memory addressing and protected mode memory addressing.

3-4	Addressing modes: Data memory addressing modes, program memory addressing modes, stack memory addressing mode.
5-6	Microprocessor Programming: Machine language, Instruction formats, Datamovement instructions, assembly language syntax, Stack manipulationinstructions, string transfer instructions, Arithmetic and logical instructions.
7-8	Program control instructions: The Jump group, different types of loops, defining function in assembly language, function call and return, introduction to interrupts.
9	Hardware Specification of 8086/8088: Pin-out diagrams of 8086/8088 microprocessors, function of pins, role of clock generator.
10	Memory Interfacing: Address decoding, interfacing of memory with 8088 and8086.
11-12	I/O Interfacing: I/O port address decoding, isolated and memory mapped I/O, interfacing of keyboard and timer, communication interface
13-14	Interrupts : Purpose of interrupts, Interrupt instructions, interrupt vectors andinterrupt descriptors, functioning of interrupt controller
15	Direct Memory Access (DMA): Basic DMA operation, functioning of DMA controller

CSE602C: Modelling and Simulation
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

UNIT-I: **5**

System models and System studies:Concept of a System, Deterministic and Stochastic Activities, Continuous and DiscreteSystems,System Modeling, Types of Models, Principles used in Modeling, CorporateModel, System Design.

UNIT-II: **5**

What is Simulation?.Technique of Simulation, The Monte-Carlo Method, Comparison of Simulation andAnalyticalMethods, Experimental nature of Simulation, Types of System Simulation, NumericalComputation Techniques for Continuous Models, Numerical Computation Techniques forDiscrete Models, Distributed Lag Models, Cobweb Models.

UNIT-III: **5**

Continuous System Simulation: Continuous System Models, Differential Equations & Applications, Feedback Systems, Simulationof an Autopilot, Interactive systems, Real Time Systems.

UNIT-IV: **5**

Concepts in Discrete Event Simulation:The Event Scheduling / Time Advance Algorithm, World Views, Manual Simulation using EventScheduling, List Processing: Lists-Basic properties and operations, Use of arrays for ListProcessing, Using Dynamic Allocation and Linked Lists.

UNIT-V: **15**

Queuing Models:Characteristics of Queuing Systems, Arrival and Service Patterns, Queue Discipline, LongRunMeasures of Performance of Queuing Systems, Time-Average Number in System, ServerUtilization, Costs in Queuing Problems, Steady State behavior of Infinite Population Markovian Models, Multiserver Queue: M/M/C/∞/∞.

UNIT-VI: **10**

Simulation software:Comparison of Simulation Packages with Programming Languages, Classification of SimulationSoftware: General Purpose vs Application Oriented Simulation Packages, Desirable SoftwareFeatures: General Capabilities, H/w and S/w Requirements, Animation and Dynamic Graphics,Statistical Capabilities, General Purpose Simulation Packages, Object Oriented Simulation, Examples of Application-oriented Simulation packages, Simulation in GPSS.

UNIT –VII: **15**

Random Number, Non-Uniform random variate Generation and Monte-Carlo Method [9 Lectures]Linear Congruential Generators, Testing Random Number Generators: Empirical and Theoreticaltests, Non-Uniform Random Variate Generator: Inverse Transform, Composition. Generating Continuous Random Variates: Uniform, Exponential, Gamma, and Normal. Generating Discrete Random Variates: Bernoulli, Binomial, Poisson. Monte-Carlo Method: Evaluation of Integral-Hit or Miss Method.

UNIT-VIII: **10**

Analysis of Simulation Data:Identifying the Distribution with Data, Types of Simulations with respect to Output Analysis,Stochastic nature of Output Data, Measures of Performance and their Estimation: Point Estimation,Confidence-Interval Estimation, Output Analysis for Terminating Simulations: StatisticalBackground, Confidence-Intervals with Specified Precision, Output Analysis forSteady StateSimulations, Variance-Reduction Technique-Antithetic variates.

UNIT-IX: **5**

Verification and Validation of Simulation Models:Model Building- Verification and Validation, Verification of Simulation Models, Calibration andValidation of Models: Validation of Model Assumptions, Validating Input-output Transformations,Input-Output Validation

References:

1. Raj Jain, Art of Computer Systems Performance Analysis, John Wiley and Sons, Inc,1991
2. Sheldon M. Ross, Simulation, 4thEd., Elsevier 2008
3. Averill M. Law and W. David Kelton, Simulation Modeling and Analysis, 3rdEd., Tata McGraw-Hill, 2003
4. Geofeerey Gordon, System Simulation, 2nd Ed., PHI, 1987
5. Jerry Banks and John S. Carson, Barry L Nelson, Discrete-Event System Simulation, 5thEd., Prentice Hall, 2010
6. Narsingh Deo, System Simulation with Digital Computers, Prentice Hall of India, 1979

CSE602C P: Modelling and Simulation(Practical)

Practical exercises based on theory

Semester- VII
Discipline Specific Elective Papers

CSE703A: Advanced Algorithms
Credit: 06
Total Marks: 100 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks)
Workload: 5 Lectures (per week), 1 Tutorial (per Week)

Course Objective

This course focuses on the study of advanced data structures and algorithms for solving problems efficiently and their theoretical behavior. The course also includes study of network flow algorithms, NP completeness and backtracking.

Course Learning Outcomes

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

1. Implement and empirically analyze advanced data-structures like tries, suffix trees.
2. Apply amortized analysis.
3. Develop more sophisticated algorithms using techniques like divide and conquer, dynamic programming, greedy strategy, and augmentation
4. Prove that certain problems are too hard to admit fast solutions.
5. Develop algorithms using backtracking for the hard problems.

Detailed Syllabus

UNIT-I: 15

Advanced Data Structures:Skip Lists, Red-Black trees, Splay Trees, Mergeable heaps (Fibonacci heaps), DS for sets - Union-Find Data Structure, Dynamic Tables, Dictionaries, Data structures for strings - Tries, Suffix trees.

UNIT-II: 10

Divide and Conquer: Counting Inversions, Closest pair of points, Integer Multiplication,

UNIT-III: 10

Greedy Algorithm: Interval Scheduling, Huffman Code, Correctness and Analysis,

UNIT-IV: 10

Dynamic Programming: Segmented Least Squares, Shortest Paths, Negative Cycles in Graphs

UNIT-V: 15

Network Flows:Max-flow problem, Ford Fulkerson Algorithm, Maximum flows and Minimum Cuts in a network, Bipartite Matching.

UNIT-VI: 15

NP Completeness:Polynomial time reductions, Efficient Certification and Definition of NP, NP Complete problems, Sequencing problems, Partitioning problems, co-NP and asymmetry of NP. Backtracking**:** Constructing All Subsets, Constructing All Permutations, Constructing All Paths in a Graph.

Tutorial

Tutorials based on Theory**.**

References

1. Cormen, T.H., Leiserson,C.E., Rivest, R.L., & Stein,C.(2010). *Introduction to Algorithms*. 3rd edition. Prentice-Hall of India Learning Pvt. Ltd.
2. Kleinberg, J., & Tardos, E. (2013). *Algorithm Design*. 1st edition. Pearson Education India.

Additional Resources

1. Basse, S., & Gleder, A. V. (1999). *Computer Algorithm – Introduction to Design and Analysis*. 3rd edition. Pearson Education.
2. Dasgupta, S., Papadimitriou,C., & Vazirani, U. (2017). *Algorithms*. 1st edition. TataMcGraw Hill.
3. Skiena, S. S. (2008). *The Algorithm Design Manual*. 2nd edition. Springer-Verlag London

Tentative weekly teaching plan is as follows:

Week	Content
1-4	Advanced Data Structures: Skip Lists, Red-Black trees, Splay Trees, Mergeable heaps (Fibonacci heaps), DS for sets - Union-Find Data Structure, Dynamic Tables, Dictionaries, Data structures for strings - Tries, Suffix trees
5	Divide and Conquer: Counting Inversions, Closest pair of points, Integer Multiplication
6-7	Greedy Algorithm: Interval Scheduling, Huffman Code, Correctness and Analysis
8-9	Dynamic Programming: Segmented Least Squares, Shortest Paths, Negative Cycles in Graphs
10-11	Network Flows: Max-flow problem, Ford Fulkerson Algorithm, Maximum flows and Minimum Cuts in a network, Bipartite Matching
12-13	NP Completeness: Polynomial time reductions, Efficient Certification andDefinition of NP, NP Complete problems, Sequencing problems, Partitioning problems, co-NP and asymmetry of NP
14-15	Backtracking: Constructing All Subsets, Constructing All Permutations,Constructing All Paths in a Graph

CSE703B: Combinatorial Optimization
Credit: 06
Total Marks: 100 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks)
Workload: 5 Lectures (per week), 1 Tutorial (per Week)

Course Objectives

This course is designed to introduce the fundamentals of combinatorial optimization to thestudents in terms of both theory and applications, so as to equip them to explore the moreadvanced areas of convex and non-convex optimizations.

Course Learning Outcomes

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

1. Model problems using linear and integer programs
2. Apply polyhedral analysis to develop algorithms for optimization problems
3. Use the concept of duality for design of algorithms

Detailed Syllabus

UNIT-I: 20

Introduction to Combinatorial Optimization Problems, Linear and Integer Programs: LPFormulation, understanding integer programs, computational complexities of IP vs LP, using LPto find optimal or approximate integral solutions, concept of integrality gap.

UNIT-II: **20**

Theory of Linear Programming and Algorithmic Perspective to Simplex Method:standard vs. equational form, basic feasible solutions, convexity and convex polyhedra, correspondence between vertices and basic feasible solutions, geometry of Simplex algorithm, exception handling (unboundedness, degeneracy, infeasibility), Simplex algorithm, avoiding cycles. **UNIT-III:** **20**

Primal-Dual Algorithms:interpretation of dual, optimality conditions for primal and dual, weak and strong duality, complementary slackness, primal-dual algorithm for the shortest path problem.

UNIT-IV: **15**

Network Flows:linear programming formulations for network flows and bipartite matching, totally unimodular matrices integral polyhedral.

Tutorials

Tutorials based on Theory

References

1. Matousek & Gartner (2007). *Understanding and Using Linear Programming*. Springer.
2. Papadimitriou, C.H. & Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and complexity*. Dover Publications.

Additional Resources:

1. Bazaraa, M.S., Jarvis, J.J., & and Sherali, H.D.(2008). *Linear Programming and Network Flows*. 2nd edition. Wiley.
2. Korte, B., & Vygen, J. (2006). *Combinatorial Optimization*. 5th edition. Springer.

Tentative weekly teaching plan is as follows:

Week	Content
1-2	Introduction to Combinatorial Optimization Problems, Linear andInteger Programs: LP Formulation, understanding integer programs,computational complexities of IP vs LP, using LP to find optimal orapproximate integral solutions, concept of integrality gap
3-6	Theory of Linear Programming and Algorithmic Perspective toSimplex Method: standard vs. equational form, basic feasiblesolutions, convexity and convex polyhedra, correspondence betweenvertices and basic feasible solutions, geometry of Simplex algorithm, exception handling (unboundedness, degeneracy,infeasibility), Simplex algorithm, avoiding cyc
7-10	Primal-Dual Algorithms: interpretation of dual, optimality conditionsfor primal and dual, weak and strong duality, complementaryslackness, primal-dual algorithm for the shortest path problem.
11-15	Network Flows: linear programming formulations for network flowsand bipartite matching, totally uni-modular matrices integralpolyhedral

CSE703C:Introduction to Data Sciences
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

UNIT-I: 15

Data Scientist's Tool Box: Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.

UNIT-II: 15

R Programming Basics: Overview of R, R data types and objects, reading and writing data, Control structures, functions, scoping rules, dates and times, Loop functions, debugging tools, Simulation, code profiling

UNIT-III: 15

Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and from colleagues in various formats. basics of data cleaning and making data "tidy".

UNIT-IV: 15

Exploratory Data Analysis: Essential exploratory techniques for summarizing data, applied before formal modelling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize highdimensional data.

UNIT-V: 15

Reproducible Research: Concepts and tools behind reporting modern data analyses in a reproducible manner, To write a document using R markdown, integrate live R code into a literate statistical program, compile R markdown documents using knitr and related tools, and organize a data analysis so that it is reproducible and accessible to others.

Reference Books

1. Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontiline" by Schroff/O'Reilly, 2013.
2. Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking" by O'Reilly, 2013.
3. John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.
4. Ian Ayres, "Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart" Ist Edition by Bantam, 2007.
5. Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1stEdition, by Wiley, 2013.
6. Matthew A. Russel, "Mining the Social Web: Data mining Facebook, Twitter, Linkedln, Goole+, GitHub, and More", Second Edition, by O'Reilly Media, 2013.

CSE703C P:Introduction to Data Sciences(Practical):

1. Write a program that prints 'Hello World' to the screen.
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
3. Write a program that prints a multiplication table for numbers up to 12.
4. Write a function that returns the largest element in a list.
5. Write a function that computes the running total of a list.
6. Write a function that tests whether a string is a palindrome.
7. Implement linear search.
8. Implement binary search.
9. Implement matrices addition , subtraction and Multiplication
10. Fifteen students were enrolled in a course. There ages were:

20 20 20 20 20 21 21 21 22 22 22 22 23 23 23

i. Find the median age of all students under 22 years

ii. Find the median age of all students

iii.	Find the mean age of all students
iv.	Find the modal age for all students
v.	Two more students enter the class. The age of both students is 23. What is now mean, mode and median ?

11. Following table gives a frequency distribution of systolic blood pressure. Compute all the measures of dispersion.

Midpoint	95.5	105.5	115.5	125.5	135.5	145.5	155.5	165.5	175.5
Number	5	8	22	27	17	9	5	5	2

12. Obtain probability distribution of , where X is number of spots showing when a six-sided symmetric die (i.e. all six faces of the die are equally likely) is rolled. Simulate random samples of sizes 40, 70 and 100 respectively and verify the frequency interpretation of probability.

13. Make visual representations of data using the base, lattice, and ggplot2 plotting systems in R, apply basic principles of data graphics to create rich analytic graphics from available datasets.

14. Use Git / Github software to create Github account. Also, create a repo using Github

CSE703D: Machine Learning

Credit: 06

Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)

Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

The course aims at introducing the basic concepts and techniques of machine learning so that astudent can apply machine learning techniques to a problem at hand.

Course Learning Outcomes

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

1. Differentiate between supervised and unsupervised learning tasks.
2. Differentiate between linear and non-linear classifiers.
3. Describe theoretical basis of SVM
4. Implement various machine learning algorithms learnt in the course.

Detailed Syllabus

UNIT-I: 15

Introduction:Basic definitions, Hypothesis space and inductive bias, Bayes optimal classifierand Bayes error, Occam's razor, Curse of dimensionality, dimensionality reduction, featurescaling, feature selection methods.

UNIT-II: 20

Regression:Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, over-fitting, regularization. Performance evaluation metrics, validation methods.

UNIT-III: 20

Classification:Decision trees, Naive Bayes classifier, k-nearest neighbor classifier, perceptron, multilayer perceptron, neural networks, back-propagation algorithm, Support Vector Machine (SVM), Kernel functions.

UNIT-IV: **20**

Clustering:Approaches for clustering, distance metrics, K-means clustering, expectation maximization, hierarchical clustering, performance evaluation metrics, validation methods.

CSE703D: Machine Learning(Practical)

For practical Labs for Machine Learning, students may use softwares likePython/Scikit-Learn. For later exercises, students can create/use their own datasets or utilize datasets from onlinerepositories like UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/).

1. Perform elementary mathematical operations like matrix - addition,multiplication, division and exponentiation using Numpy
2. Perform elementary logical operations (like OR, AND, Checking forEquality, NOT, XOR).
3. Create, initialize and display simple variables and simple strings and use simple formatting for variable.
4. Create/Define single dimension / multi-dimension arrays, and arrays with specific values likearray of all ones, all zeros, array with random values within a range, or a diagonal matrix.
5. Use command to compute the size of a matrix, size/length of a particular row/column, loaddata from a text file, store matrix data to a text file, finding out variables and their features in thecurrent scope.
6. Perform basic operations on matrices (like addition, subtraction, multiplication) and displayspecific rows or columns of the matrix.
7. Perform other matrix operations like converting matrix data to absolute values, taking thenegative of matrix values, adding/removing rows/columns from a matrix, finding the maximumor minimum values in a matrix or in a row/column, and finding the sum of some/allelements in a matrix.
8. Create various type of plots/charts like histograms, plot based on sine/cosine function based ondata from a matrix. Further label different axes in a plot and data in a plot.
9. Generate different subplots from a given plot and color plot data.
10. Use conditional statements and different type of loops based on simple example/s.
11. Perform vectorized implementation of simple matrix operation like finding the transpose of amatrix, adding, subtracting or multiplying two matrices.
12. Implement Linear Regression problem. For example, based on a dataset comprising ofexisting set of prices and area/size of the houses, predict the estimated price of a given house.
13. Based on multiple features/variables perform Linear Regression. For example, based on anumber of additional features like number of bedrooms, servant room, number of balconies,number of houses of years a house has been built – predict the price of a house.
14. Implement a classification/ logistic regression problem. For example based on differentfeatures of students data, classify, whether a student is suitable for a particular activity. Based onthe available dataset, a student can also implement another classification problem like checkingwhether an email is spam or not.
15. Use some function for regularization of dataset based on problem 14.
16. Use some function for neural networks, like Stochastic Gradient Descent or backpropagation-algorithm to predict the value of a variable based on the dataset of problem 14.
17. Implement logistic regression classification with (a) gradient descent and (b) stochastic gradient descent method. Plot cost function over iteration.
18. Experiment with logistic regression by adding momentum term, and adaptive subgradient method
19. Write the code to learn weights of a perceptron for Boolean functions (NOT, OR, AND, NOR, and NAND).
20. Implement a feed-forward neural network for solving (a) regression and (b) 2-class classification problem. Also experiment with hyper-parameter tuning.

21. Train and test a feed-forward neural network for multi-class classification using softmax layer as output.

References

1. Flach, P. (2015). *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press.
2. Mitchell, T.M. (2017). *Machine Learning*. McGraw Hill Education.

Additional References:

1. Christopher & Bishop, M. (2016). *Pattern Recognition and Machine Learning*. New York: Springer-Verlag
94
2. Haykins, S.O. (2010). *Neural Networks and Learning Machines*. 3rd edition. PHI.

Tentative weekly teaching plan is as follows:

Week	Content
1	Basic definitions, Hypothesis space and inductive bias, Bayes optimal classifier and Bayes error, Occam's razor
2	Curse of dimensionality, dimensionality reduction, feature scaling, featureselection methods
3	Linear regression with one variable, linear regression with multiple variables
4 -5	Gradient descent, logistic regression, over-fitting, regularization
6	Performance evaluation metrics, validation methods.
7	Decision trees, Naive Bayes classifier ,k-nearest neighbor classifier
8 - 9	Perceptron, Multilayer perceptron, neural networks, back-propagation algorithm
10-11	Support Vector Machine (SVM), Kernel functions
12	Approaches for clustering, distance metrics
13	K-means clustering, expectation maximization
14	Hierarchical clustering
15	Clustering validation methods, performance evaluation metrics

Semester- VII
Discipline Specific Elective Papers

CSE804A: Deep Learning
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective
The objective of this course is to introduce students to deep learning algorithms and their applications in order to solve real problems.

Course Learning Outcomes
On successful completion of this course, the student will be able to:

1. Describe the feed-forward and deep networks.
2. Design single and multi-layer feed-forward deep networks and tune various hyper-parameters.
3. Implement deep neural networks to solve a problem
4. Analyse performance of deep networks.

Detailed Syllabus

UNIT-I: **10**

Introduction:Historical context and motivation for deep learning; basic supervisedclassification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive sub-gradient method.

UNIT-II: **10**

Neural Networks:Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyper parameter tuning.

UNIT-III: **10**

Convolution Neural Networks:Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.

UNIT-IV: **15**

Sequence Modeling:Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks, LSTM networks.

UNIT-V: **15**

Autoencoders:Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

UNIT-VI: **15**

Structuring Machine Learning Projects:Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labeled data, bias and variance with mismatched data distributions, transfer learning, multi-task learning.

CSE804A P: Deep Learning(Practical)

1. Implement logistic regression classification with (a) gradient descent and (b) stochastic gradient descent method. Plot cost function over iteration.

2. Experiment with logistic regression by adding momentum term, and adaptive subgradient method
3. Write the code to learn weights of a perceptron for Boolean functions (NOT, OR, AND, NOR, and NAND).
4. Implement a feed-forward neural network for solving (a) regression and (b) 2-class classification problem. Also experiment with hyper-parameter tuning.
5. Train and test a feed-forward neural network for multi-class classification using softmax layer as output.
6. Create a 2D and 3D CNN for image classification. Experiment with different depth of network, striding and pooling values.
7. Implement (a) RNN for image classification, (b) GRU network and (c) Implement LSTM networks
8. Implement an auto-encoder, denoising autoencoders and sparse autoencoders.
9. Design a stochastic encoders and decoders.

References:

1. Bunduma, N. (2017). *Fundamentals of Deep Learning*. O'reilly Books.
2. Heaton, J.(2015). *Deep Learning and Neural Networks*, Heaton Research Inc.

Additional References:

1. Goodfellow, I. (2016). *Deep Learning*. MIT Press.
2. Deng, L., & Yu, D. (2009). *Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing).* Publishers Inc.
3. Hall, M.L, (2011). *Deep Learning*. VDM Verlag

Tentative weekly teaching plan is as follows:

Week	Content
1	Introduction: Historical context and motivation for deep learning;basic supervised classification task
2-3	Optimizing logistic classifier using gradient descent, stochasticgradient descent, momentum, and adaptive sub-gradient method
4-5	Neural Networks: Feedforward neural networks, deep networks,regularizing a deep network, model exploration, and hyperparameter tuning
6-7	Convolution Neural Networks: Introduction to convolution neuralnetworks: stacking, striding and pooling, applications like image,and text classification
8	Sequence Modeling: Recurrent Nets: Unfolding computationalgraphs, recurrent neural networks (RNNs), bidirectional RNNs
9	Encoder-decoder sequence to sequence architectures, deep recurrentnetworks, LSTM networks
10	Autoencoders: Undercomplete autoencoders, regularizedautoencoders, sparse autoencoders
11-12	Denoising autoencoders, representational power, layer, size, anddepth of autoencoders, stochastic encoders and decoders.
13	Structuring Machine Learning Projects: Orthogonalization,evaluation metrics, train/dev/test distributions,

14-15	Size of the dev and test sets, cleaning up incorrectly labeled data,bias and variance with mismatched data distributions, transferlearning, multi-task learning.

CSE804B: Unix Network Programming
Credit: 06
Total Marks: 150 Marks (Theory: 75 Marks, Internal Assessment: 25 Marks, Practical:50 Marks)
Workload: 4 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces the concepts of Internet protocols, ports used during communication,Client/Server concepts and various transport protocols used in computer network applicationsand services. The objective is to equip the students with technical knowledge of it comprises ofthe study of the sockets used with TCP and UDP include IPV4 & IPV6.

Course Learning Outcomes

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

1. Describe and analyse the various Internet Transport layer protocols used in TCP/IP ANDUDP.
2. Comprehend the concepts and structures of both TCP based connection-oriented and UDP based connection-less client server applications.
3. Write various real-life client-server applications using socket programming.
4. Modify, maintain and extend the present internet client-server applications and write any new type of internet applications to suit the current needs of Internet users.

Detailed Syllabus

UNIT-I: **10**

Introduction:Basics of Client Server applications, Example of day time client server,concurrent servers, protocols, sockets, port numbers.

UNIT-II: **15**

Connection-oriented and Connection-less client server Applications:Elementary TCPsockets – Socket, connect, bind, listen, accept, fork and exec function, close function, SocketAddress Structures, Byte Ordering and Manipulation Functions, TCP Client and Server for Echo,Signal Handling in case of crashing and rebooting of server, Shutdown process function.

UNIT-III: **10**

Socket Options:Getsockopt and stockpot functions, Socket states, Generic socket option

UNIT-IV: **15**

Connection-oriented and connection-less Sockets:TCP-oriented basic concurrent client server applications, UDP oriented Echo client and server application, Handling of errors like lost datagram, Lack of flow control with UDP, determining outgoing interface with UDP.

UNIT-V: **15**

Elementary name and Address conversions:Domain Name System, socket functions like gethostbyname, gethostbyname2, gethostbyaddr function, uname function, gethostname function, getservbyname and getservbyport functions.

UNIT-VI: **10**

Advanced Sockets:Daemon Processes, Multithreaded server, Raw sockets.

CSE804B(P): Unix Network Programming(Practical)

1. Implement TCP Echo client and TCP Echo server (Iterative).
2. Implement TCP Echo client and TCP Echo server (Concurrent).
3. Implement TCP daytime client and TCP daytime server (Iterative).
4. Implement TCP daytime client and TCP daytime server (concurrent).
5. Implement UDP Echo Client and UDP Echo Server.
6. Implement UDP daytime Client and UDP daytime server.
7. Implement TCP client and server (concurrent) where client gets input from the user andsends it to server.

Server displays it on the screen. Server then gets another input fromthe user and sends it to client. Client displays it on the screen. The process continues tillserver or client sends "bye" to the other party.

8. Implement TCP client and server (concurrent) where client requests server to transfer a file. Assume file is smaller than 1K size. If the file is present on the server, it is sent to the client otherwise an error message is sent to client. Client copies the file on the hard disk and disconnects.
9. Implement UDP client and UDP server where server displays the IP address and port number of the client sending the datagram. Client sends a datagram (size 64 bytes) three times to the same server. Server sends the message back to client. Client reports the time elapsed in sending and receiving of the message. Use connected UDP sockets.

10. Write to program to
    1. display name of the host
    2. all IP addresses of the host.
    3. Check whether FTP and HTTP services are running on the system.
    4. Display the name of the service running on port number specified by user.

References

1. Stevens, R. W., Fenner, B., & Rudoff, A. M. (2010). *Unix Network Programming: The Sockets Networking API*. 3rd edition. PHI.

Additional Resources:

1. Forouzan, B. A. (2017). *Data Communication and Networking. 4th e*dition. McGraw-Hill Education.
2. Stevens, R. W. (2009). *Unix Network Programming*. 1st edition. PHI.
3. Tanenbaum, A. S. (2012). *Computer Networks*. 5th edition. Pearson Education

Tentative weekly teaching plan is as follows:

Week	Contents
1	Introduction, client server applications, protocols, port numbers
2-3	Sockets Functions, fork and exec function, Socket addressstructure
4	TCP Echo Server
5	Signal Handling
6-7	I/O Multiplexing
8-9	Socket Options, Getsockopt and stockpot functions, socketstates, generic socket options
10	Elementary UDP sockets, TCP and UDP oriented client serverapplications
11	Elementary name and Address conversions, DNS, socketfunctions
12	Daemon Processes
13-14	Multithreaded server
15	Raw sockets

CSE804C: Project Work / Dissertation
Credit: 06
Total Marks: 100 Marks

Course Objective

The students will undergo one semester of project work based on the concepts studied in asubject of their choice. The objective is to train the students for the industry by exposing them toprototype development of real life software.

Course Learning Outcomes

On successful completion of this course, a student will be able to:

1. develop a project plan based on informal description of the project.
2. implement the project as a team.
3. write a report on the project work carried out by the team and defend the work done bythe team collectively.
4. present the work done by the team to the evaluation committee.

UNIT-I:

The students will work on any project based on the concepts studied in core/elective/ skill based elective courses. Specifically, the project could be a research study, or a software developmentproject.

UNIT-II:

Project Group Organization/Plan

- Students will initially prepare a synopsis (500 words) and submit it to their respective department.
- For a given project, the group size could be a maximum of four (04) students.
- Each group will be assigned a teacher as a supervisor who will be responsible for their lab classes.
- A maximum of four (04) projects would be assigned to one teacher.

UNIT-III:

Project Evaluation

- 100 marks for end semester examination comprising Viva/presentation (50 marks) and project report evaluation (50 marks): to be awarded jointly by the examiner and supervisor / mentor.
- 50 marks for continuous evaluation (to be awarded by the supervisor/mentor). Work carried out in each lab session will be assessed out of five marks (zero for being absent). Finally, the marks obtained will be scaled out of a maximum of 50 marks. For example, if 30 lab sessions are held in a semester, and a student has obtained an aggregate of 110 marks, then he/she will be assigned round (110/(30*5)) i.e. 37 marks.
- The students will submit only the soft copies of the report.
- The reports may be retained by the examiners.

Practical

Practical/discussion sessions based on the area of the project.

Skill Enhancement Course

CSSE101A:Web Design and development

Credit: 04

Total Marks: 100 Marks (Theory: 35 Marks, Internal Assessment: 15 Marks, Practical:50 Marks)

Workload: 2 Lectures (per week), 4 Practical(per Week)

Course Objective

This course will introduce students to the fundamental concepts of web development. Thiscourse will equip students with the ability to design and develop a dynamic website usingtechnologies like HTML, CSS, JavaScript, PHP and MySQL on platform likeWAMP/XAMP/LAMP.

Course Learning Outcomes

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

1. Design and develop a website
2. Use Front end technologies like HTML, CSS and JavaScript
3. Use backend technologies like PHP and MySQL
4. Work on platforms like WAMP/XAMP/LAMP

Detailed Syllabus

UNIT-I: 5

Introduction to Static and Dynamic Websites (Website Designing and Anatomy of Webpage)

UNIT-II: 5

Introduction to HTML and CSS (Basic Tags, Lists, Handling Graphics, Tables, Linking, Frames, Forms), Introduction to DOM

UNIT-III: 5

Introduction to JavaScript (Basic Programming Techniques & Constructs, GET/POST Methods, Operators, Functions, DOM Event handling, Forms Validation, Cookies), Inter-pagecommunication and form data handling using JavaScript

UNIT-IV: 10

Introduction to PHP (Working, Difference with other technologies like JSP and ASP), PHPProgramming Techniques (Data types, Operators, Arrays, Loops, Conditional statements,Functions, Regular expressions)

UNIT-V: 10

Form Data Handling with PHP, Database connectivity and handling using PHP-MySQL

CSSE101A(P):Web Design and development (Practical)

1. Practicals based on HTML
2. Practicals based on CSS
3. Practicals based on PHP
4. Practicals to create HTML forms
5. Practicals based on database connectivity with

References:

1. Bayross, I. (2013). *Web enabled commercial application development using HTML, JavaScript, DHTML and PHP*. 4th edition. BPB Publication.

2. Holzner, S.(2007). *PHP: The Complete Reference Paperback*, McGraw Hill Education (India).

Additional Resources

1. Boronczyk, T., & Psinas, M. E. (2008). *PHP and MYSQL (Create-Modify-Reuse).* Wiley India Private Limited.

2. Welling, L., & Thompson, L. (2008). *PHP and MySQL Web Development*. 4th edition. Addition Paperback, Addison-Wesley Professional.

3. Nixon, R. (2014). *Learning PHP, MySQL, JavaScript, CSS & HTML5*. 3rd edition. Paperback, O'reilly Media
4. Sklar, D., & Trachtenberg, A., (2014). *PHP Cookbook: Solutions & Examples for PHP Programmers.* 2nd edition. O'reilly Media

CSSE101B:Linux/Unix Programming
Credit: 04
Total Marks: 100 Marks (Theory: 35 Marks, Internal Assessment: 15 Marks, Practical:50 Marks)
Workload: 2 Lectures (per week), 4 Practical(per Week)

UNIT-I: 10

Introduction:What is linux/unix Operating systemsDifference between linux/unix and other operating systemsFeatures and ArchitectureVarious Distributions available in the marketInstallation, Booting and shutdown processSystem processes (an overview)External and internal commands Creation of partitions in OSProcesses and its creation phases – Fork, Exec, wait

UNIT-II: 10

User Management and the File System:Types of Users, Creating users, Granting rightsUser management commandsFile quota and various file systems availableFile System Management and Layout, File permissionsLogin process, Managing Disk QuotasLinks (hard links, symbolic links)

UNIT-III: 15

Shell introduction and Shell Scripting:What is shell and various type of shell, Various editors present in linuxDifferent modes of operation in vi editorWhat is shell script, Writing and executing the shell scriptShell variable (user defined and system variables)System calls, Using system calls pipes and FiltersDecision making in Shell Scripts (If else, switch), Loops in shellFunctions Utility programs (cut, paste, join, tr , uniq utilities)Pattern matching utility (grep)

Reference Books:

1. Sumitabha, Das, Unix Concepts And Applications, Tata McGraw-Hill Education, 2006
2. Michael Jang RHCSA/ RHCE Red Hat Linux Certification: Exams (Ex200 & Ex300) (Certification Press), 2011
3. Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education, 2nd Edition ,2010
4. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking
API, Vol. 1, 3rd Edition,2014

CSSE101B(P):Linux/Unix Programming:

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify ―cal‖ command to display calendars of the specified months.
3. Write a shell script to modify ―cal‖ command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message – ―Entered login name is invalid‖.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of ―who‖ command along with the total number of users .
7. Write a shell script to display the multiplication table any number,

8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number.
10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
11. Write a shell script to find the LCD(least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the binomial coefficient C(n , x).
15. Write a shell script to find the permutation P(n,x).
16. Write a shell script to find the greatest number among the three numbers.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.

CSSE101C: Office Automation Tools
Credit: 04
Total Marks: 100 Marks (Theory: 35 Marks, Internal Assessment: 15 Marks, Practical: 50 Marks)
Workload: 2 Lectures (per week), 4 Practical(per Week)

UNIT-I: **5**

Introduction to open office/MS office/Libre office

UNIT-II: **10**

Word Processing: Formatting Text, Pages, Lists, Tables

UNIT-III; **10**

Spreadsheets: Worksheets, Formatting data, creating charts and graphs, using formulas and functions, macros, Pivot Table

UNIT-IV: **10**

Presentation Tools: Adding and formatting text, pictures, graphic objects, including charts, objects, formatting slides, notes, hand-outs, slide shows, using transitions, animations

Books Recommended:

1. Sushila Madan , Introduction to Essential tools,JBA,2009.
2. Anita Goel, Computer Fundamentals, Pearson, 2012

CSSE101C(P): Office Automation Tools(Practical):
Practical List for WORD:

1. Create a **telephone directory**.
   - The heading should be 16-point Arial Font in bold
   - The rest of the document should use 10-point font size
   - Other headings should use 10-point Courier New Font.
   - The footer should show the page number as well as the date last updated.
2. Design a time-table form for your college.
   - The first line should mention the name of the college in 16-point Arial Font and should be bold.
   - The second line should give the course name/teacher's name and the department in 14-point Arial.
   - Leave a gap of 12-points.
   - The rest of the document should use 10-point Times New Roman font.
   - The footer should contain your specifications as the designer and date of creation.

3. Create the following one page documents.

(a) Compose a note inviting friends to a get-together at your house, including a list of things to bring with them.

(b) Design a certificate in landscape orientation with a border around the document.

4. Create the following document: A newsletter with a headline and 2 columns in portrait orientation, including at least one image surrounded by text.

5. Convert following text to a table, using comma as delimiter

Type the following as shown (do not bold).

Color, Style, Item
Blue, A980, Van
Red, X023, Car
Green, YL724, Truck
Name, Age, Sex
Bob, 23, M
Linda, 46, F
Tom, 29, M

6. Prepare a grocery list having four columns (Serial number, the name of the product, quantity and price) for the month of April, 06.

- Font specifications for Title (Grocery List): 14-point Arial font in bold and italics.
- The headings of the columns should be in 12-point and bold.
- The rest of the document should be in 10-point Times New Roman.
- Leave a gap of 12-points after the title.

7. XYZ Publications plans to release a new book designed as per your syllabus. Design the first page of the book as per the given specifications.

(a) The title of the book should appear in bold using 20-point Arial font.

(b) The name of the author and his qualifications should be in the center of the page in 16-point Arial font.

(c) At the bottom of the document should be the name of the publisher and address in 16-point Times New Roman.

(d) The details of the offices of the publisher (only location) should appear in the footer.

8. Create the following one page documents.

(a) Design a Garage Sale sign.

(b) Make a sign outlining your rules for your bedroom at home, using a numberedlist.

9.Enter the following data into a table given on the next page.

Salesperson	**Dolls**	**Trucks**	**Puzzles**
John	1327	1423	1193
Stephen	1421	3863	2934
victoria	5214	3247	5467
Nishi	2190	1278	1928
Davidson	1201	2528	1203
Santosh	4098	3079	2067

Add a column Region (values: S, N, N, S, S, S) between the Salesperson and Dolls columns to the given table Sort your table data by Region and within Region by Salesperson in ascending order:

Practical List for EXCEL

Q1. Consider the following employee worksheet:-
Gross = Basic + HRA + VA
Net = Gross –PF
PF is 8% for all Grades
VA is 15000, 10000 and 7000 for Grades 1, 2 and 3.
i) Find max, min and average salary of employees in respective Grade

ii)	Count no. of people where VA>HRA
iii)	Find out most frequently occurring grade.

iv)	Extract records where employee name starts with "A" has HRA>10000
v)	Print Grade wise report of all employees with subtotals of net salary and also grand totals. Use subtotal command. Extract records where Grade is 1 or 2 and salary is between 10000 and 20000 both inclusive.

Q2. In a meeting of a marketing department of an organization it has been decided that price of selling an item is fixed at Rs40. It was resolved to increases the sell of more of more items and getting the profit of Rs40,000/.Use Goal Seek of find out how many items you will have to sell to meet your profit figure.

Q3.To study the variation in volume with pressure for a sample of an air at constanttemperature by plotting a graph for P – V and P-I/V. Sample observations are:-

Pressure(P)	Volume (V)	I/V	PV	P/V
75	20			
78.9	19			
83.3	18			
88.2	17			

Q4. Plot the chart for marks obtained by the students (out of 5) vs. frequency (total number of students in class is 50).

Q5. Create the following worksheet(s) containing an year wise sale figure of five salesmen inRs.

Salesman	2002	2003	2004	2005
Stephen	10000	12000	20000	50000
Victoria	15000	18000	50000	60000
Hoshika	20000	22000	70000	70000
Delina	30000	30000	100000	80000
Martin	40000	45000	125000	90000

Apply the following Mathematical & Statistical functions:

i) Calculate the commission for each salesman under the condition :-

a) If total sales is greater than Rs. 3, 00,000/-, then commission is 10% of total sale made by the salesman.

b) Otherwise, 4% of total sale.

ii) Calculate the maximum sale made by each salesman.

iii) Calculate the maximum sale made in each year.

iv) Calculate the minimum sale made by each salesman.

v) Calculate the minimum sale made in each year.

vi) Count the no. of sales persons.

vii)Calculate the cube of sales made by Mohan in the year 2002.

viii) Find the difference in sales by salesman Mitra between the year 2002 and 2003. Find the absolute value of difference.

ix) Also calculate the Mode, Stddev, Variance, Median for the sale made by each salesman.

ix) Calculate the year wise Correlation coefficient between the sales man Mohan and Mitra

Q6. Consider the following worksheet for APS 1st year students:-

S.No.	Name	PH	CH	BY	MT	CS	Total	%	Grade
Marks									
1									
2									
Grade is calculated as follows:-									
If % >=90	Grade A								

If % >=80 &<90	Grade B								
If % >=70 &<80	Grade C								
If % >=60 &<70	Grade D								

Otherwise students will be declared fail.

i) Calculate Grade using if function

ii) Sort the data according to total marks

iii) Apply filter to display the marks of the students having more than 65% marks.

iv) Draw a pie chart showing % marks scored in each subject by the topper of the class.

v) Draw the doughnut chart of the data as in (iv)

vi) Enter the S.No. of a student and find out the Grade of the student using VLOOKUP.

vii)Extract all records where name

a) Begins with "A"

b) Contains "A"

c) Ends with "A"

Practical List for Power Point:

1. Create five Power point slides. Each slide should support different format. In theseslides explain areas of applications of IT. Make slide transition time as 10 seconds.
2. Create five Power Point slides to give advantages/disadvantages of computer,application of computers and logical structure of computer.
3. Create five Power Point slides detailing the process of internal assessment. It should be aself-running demo.

CSSE202A: Programming in Python

Credit: 04

Total Marks: 100 Marks (Theory: 35 Marks, Internal Assessment: 15 Marks, Practical: 50 Marks)

Workload: 2 Lectures (per week), 4 Practical(per Week)

Course Objective

This course is designed to introduce the student to the basics of programming using Python. Thecourse covers the topics essential for developing well documented modular programs usingdifferent instructions and built-in data structures available in Python.

Course Learning Outcomes

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

1. Develop, document, and debug modular python programs to solve computational problems.
2. Select a suitable programming construct and data structure for a situation.
3. Use built-in strings, lists, sets, tuples and dictionary in applications.
4. Define classes and use them in applications.
5. Use files for I/O operations.

Detailed Syllabus

UNIT-I: 5

Introduction to Programming using Python**:** Structure of a Python Program, Functions,Interpreter shell, Indentation. Identifiers and keywords, Literals, Strings, Basic operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment Operator,Bit wise operator).

UNIT-II: **5**

Building blocks of Python:Standard libraries in Python, notion of class, object and method.

UNIT-III: **5**

Creating Python Programs:Input and Output Statements, Control statements:-branching, looping, Exit function, break, continue and pass, mutable and immutable structures. Testing and debugging a program

UNIT-IV: **10**

Built-in data structures:Strings, lists, Sets, Tuples and Dictionary and associated operations. Basic searching and sorting methods using iteration and recursion.

UNIT-V: **5**

Visualization using 2D and 3D graphics**:** Visualization using graphical objects like Point, Line, Histogram, Sine and Cosine Curve, 3D objects

UNIT-VI: **5**

Exception Handling and File Handling**:** Reading and writing text and structured files, Errors and Exceptions.

CSSE202A(P): Programming in Python(Practical)

Practical

1. Execution of expressions involving arithmetic, relational, logical, and bitwise operators
107
in the shell window of Python IDLE.

2. Write a Python function to produce the outputs such as:

a)

```
*
* * *
* * * * *
* * *
*
```

(b)

```
1
232
34543
4567654
567898765
```

3. Write a Python program to illustrate the various functions of the “Math” module.

4. Write a function that takes the lengths of three sides: **side1**, **side2** and **side3** of thetriangle as the input from the user using **input** function and return the area of thetriangle as the output. Also, assert that sum of the length of any two sides is greaterthan the third side.

5. Consider a showroom of electronic products, where there are various salesmen. Eachsalesman is given a commission of 5%, depending on the sales made per month. In casethe sale done is less than 50000, then the salesman is not given any commission. Writea function to calculate total sales of a salesman in a month, commission and remarks

for the salesman. Sales done by each salesman per week is to be provided as input.Assign remarks according to the following criteria**:**

Excellent: Sales >=80000
Good: Sales>=60000 and <80000
Average: Sales>=40000 and <60000
Work Hard: Sales < 40000

6. Write a Python function that takes a number as an input from the user and computes its factorial.
7. Write a Python function to return nth terms of Fibonacci sequence
8. Write a function that takes a number with two or more digits as an input and finds itsreverse and computes the sum of its digits.
9. Write a function that takes two numbers as input parameters and returns their leastcommon multiple and highest common factor.
10. Write a function that takes a number as an input and determine whether it is prime ornot.
11. Write a function that finds the sum of the n terms of the following series:

a) 1 – x2 /2! + x4 /4! – x6 /6! + … xn /n!

b) 1 + x2 /2! + x4 /4! + x6 /6! + … xn /n!

12. Write a Python function that takes two strings as an input from the user and counts thenumber of matching characters in the given pair of strings.
13. Write a Python function that takes a string as an input from the user and displays itsreverse.
14. Write a Python function that takes a string as an input from the user and determineswhether it is palindrome or not.
15. Write a Python function to calculate the sum and product of two compatible matrices
16. Write a function that takes a list of numbers as input from the user and produces thecorresponding cumulative list where each element in the list present at index i is thesum of elements at index j <= i.
17. Write a function that takes **n** as an input and creates a list of n lists such that ith listcontains first five multiples of i.
18. Write a function that takes a sentence as input from the user and calculates thefrequency of each letter. Use a variable of dictionary type to maintain the count.
19. Write a Python function that takes a dictionary of *word***:***meaning* pairs as an input fromthe user and creates an inverted dictionary of the form meaning:list-of-words.
20. Usage of Python debugger tool-pydb and PythonTutor.
21. Implementation of Linear and binary search techniques
22. Implementation of selection sort, insertion sort, and bubble sort techniques
23. Write a menu-driven program to create mathematical 3D objectsCurve, Sphere, Cone, Arrow, Ring, Cylinder.
24. Write a program that makes use of a function to accept a list of n integers anddisplays a histogram.
25. Write a program that makes use of a function to display sine, cosine, polynomial andexponential curves.
26. Write a program that makes use of a function to plot a graph of people with pulse ratep vs. height h. The values of p and h are to be entered by the user.
27. Write a function that reads a file **file1** and displays the number of words and thenumber of vowels in the file.
28. Write a Python function that copies the content of one file to another.
29. Write a function that reads a file **file1** and copies only alternative lines to another file**file2**. Alternative lines copied should be the odd numbered lines.

References

1. Downey, A.B., (2015), *Think Python–How to think like a Computer Scientist*, 3rd edition. O'Reilly Media.

2. Taneja, S. & Kumar, N., (2017), *Python Programming- A Modular Approach*. Pearson Education.

Additional Resources

1. Brown, M. C. (2001). *The Complete Reference: Python*, McGraw Hill Education.
2. Dromey, R. G. (2006), *How to Solve it by Computer*, Pearson Education.
3. Guttag, J.V.(2016), *Introduction to computation and programming using Python*. MIT Press.
4. Liang, Y.D. (2013), *Introduction to programming using Python*. Pearson Education

Tentative weekly teaching plan is as follows:

Week	Content
1.	Python Programming: An Introduction Structure of a Python program, understanding Python interpreter/Pythonshell, indentation. Atoms, identifiers and keywords, literals, Pythonstrings, arithmetic operator, relational operator, logical or Boolean operator, bit wise operators.
2	Variables and Functions Python standard libraries such as sys and math. Variables andassignment statements. Built-in functions such as input and print.
3-4	Control Structures if conditional statement and for loop, While loop, break, continue, andpass statement, else statement. Infinite loop
5	Functions Function definition and call, default parameter values, keywordarguments, assert statement
6	Strings and Lists Strings-slicing, membership, and built-in functions on stringsLists- list operations.
7.	Mutable object Lists- built-in functions, list comprehension, passing list as arguments,copying list objects.
8	Sets, tuples, and dictionary- associated operations and built-in functions.
9	Testing and Debugging Determining test cases, use of python debugger tool- pydb fordebugging
10	Searching and Sorting Linear search, binary search, selection sort, insertion sort, and bubblesort
11	Python 2D and 3D Graphics Visualization using graphical objects like point, line, histogram, sineand cosine curve, 3D objects
12	File Handling Reading and writing text and structured files.
13	Errors and Exceptions Types of errors and exceptions, and exception handling
14	Classes Notion of class, object, and method.

CSSE202B: Introduction to R Programming
Credit: 04
Total Marks: 100 Marks (Theory: 35 Marks, Internal Assessment: 15 Marks, Practical: 50 Marks)
Workload: 2 Lectures (per week), 4 Practical(per Week)

Course Objective

This course introduces R, which is a popular statistical programming language. The coursecovers data reading and its manipulation using R, which is widely used for data analysisinternationally. The course also covers different control structures and design of user-definedfunctions. Loading, installing and building packages are covered.

Course Learning Outcomes

On successful completion of the course, students will be able to do following:
1. Develop an R script and execute it
2. Install, load and deploy the required packages, and build new packages for sharing and reusability
3. Extract data from different sources using API and use it for data analysis
4. Visualize and summarize the data
5. Design application with database connectivity for data analysis

Detailed Syllabus

UNIT-I: 10

Introduction:R interpreter, Introduction to major R data structures like vectors, matrices,arrays, list and data frames, Control Structures, vectorized if and multiple selection, functions. **UNIT-II: 10**

Installing, loading and using packages:Read/write data from/in files, extracting data fromweb-sites, Clean data, Transform data by sorting, adding/removing new/existing columns,centring, scaling and normalizing the data values, converting types of values, using string in-builtfunctions, Statistical analysis of data for summarizing and understanding data, Visualizing datausing scatter plot, line plot, bar chart, histogram and box plot

UNIT-III 10

Designing GUI:Building interactive application and connecting it with database.

UNIT-IV 5

Building Packages.

CSSE202B(P): Introduction to R Programming(Practical)

Q1. Write an R script to do the following:
a) simulate a sample of 100 random data points from a normal distribution with mean 100 andstandard deviation 5 and store the result in a vector.
b) visualize the vector created above using different plots.
c) test the hypothesis that the mean equals 100.
d) use wilcox test to test the hypothesis that mean equals 90.
Q2. Using the Algae data set from package DMwR to complete the following tasks.
a) create a graph that you find adequate to show the distribution of the values of algae a6.
b) show the distribution of the values of size 3.
c) check visually if oPO4 follows a normal distribution.
d) produce a graph that allows you to understand how the values of NO3 are distributed across the sizes of river.
e) using a graph check if the distribution of algae a1 varies with the speed of the river.
f) visualize the relationship between the frequencies of algae a1 and a6. Give the

appropriate graph title, x-axis and y-axis title.

Q3. Read the file Coweeta.CSV and write an R script to do the following:

a) count the number of observations per species.

b) take a subset of the data including only those species with at least 10 observations.

c) make a scatter plot of biomass versus height, with the symbol colour varying by species, and use filled squares for the symbols. Also add a title to the plot, in italics.

d) log-transform biomass, and redraw the plot.

Q4. The built-in data set mammals contain data on body weight versus brain weight. Write R commands to:

a) Find the Pearson and Spearman correlation coefficients. Are they similar?

b) Plot the data using the plot command .

c) Plot the logarithm (log) of each variable and see if that makes a difference.

Q5. In the library MASS is a dataset UScereal which contains information about popular breakfast cereals. Attach the data set and use different kinds of plots to investigate the following relationships:

a) relationship between manufacturer and shelf

b) relationship between fat and vitamins

c) relationship between fat and shelf

d) relationship between carbohydrates and sugars

e) relationship between fibre and manufacturer

f) relationship between sodium and sugars

Q6. Write R script to:

a) Do two simulations of a binomial number with n = 100 and p = .5. Do you get the same results each time? What is different? What is similar?

b) Do a simulation of the normal two times. Once with n = 10, $\mu = 10$ and $\sigma = 10$, the other with n = 10, $\mu = 100$ and $\sigma = 100$. How are they different? How are they similar? Are both approximately normal?

Q7. Create a database medicines that contains the details about medicines such as {manufacturer, composition, price}. Create an interactive application using which the user can find an alternative to a given medicine with the same composition.

Q8. Create a database songs that contains the fields {song_name, mood, online_link_play_song}. Create an application where the mood of the user is given as input and the list of songs corresponding to that mood appears as the output. The user can listen to any song form the list via the online link given.

Q9. Create a package in R to perform certain basic statistics functions.

Mini project using data set of your choice from Open Data Portal (https://data.gov.in/) for the following exercises

References

1. Cotton, R., *Learning R: a step by step function guide to data analysis.* 1st edition. O'reilly Media Inc.

Additional Resources:

2. Gardener, M.(2017). *Beginning R: The statistical programming language*, WILEY.

3. Lawrence, M., & Verzani, J. (2016). *Programming Graphical User Interfaces in R*. CRC press. (ebook)

Tentative weekly teaching plan is as follows:

Week	**Content**
1	R interpreter, Introduction to major R data structures like vectors, matrices, arrays, list and data frames
2	Flow control and loops, looping over list and array
3	User-defined functions
4	Installing, loading different packages for file handling
5	Reading and writing files of different formats using inbuilt packages
6	Using inbuilt packages for data cleaning
7	Transformation of data for statistical analysis
8	Exploring and summarizing data using statistical methods: mean, median, mode
9	Exploring and summarizing data using statistical methods: quantiles, Building contingency table
10	Data visualization using Scatter Plot, line graph, histogram, barchart, boxplot
11	Designing GUI
12	Continuing with creating GUI for application, building package
13-14	Using inbuilt packages for database connectivity
15	Building complete application with GUI and database connectivity