Department of Information Technology

Syllabus

for

Master in Computer Applications

(Three Year Course)

Batch: 2021 onwards

Sardar Beant Singh State University, Gurdaspur

Master of Computer Applications (MCA):

It is a Post Graduate (PG) Programme of 3 years duration (6 semesters)

Eligibility: Passed BCA/B.Sc(CS/IT)/B.Voc with Computer as a major subject/Bachelor's Degree in CSE/IT or equivalent degree of minimum three years duration.

Or

Passed B.Sc./ B.Com./ B.A. with Mathematics at l0+2 Level or at Graduation Level.

(Guresh Pal Singh) (Anil)

Syllabus Scheme of Master in Computer Applications (MCA)

Batch: 2021

Sr. No.	Subject Code	Subject	Credits			Marks Distribution		Total Marks
		Semester I	**L**	**T**	**P**	**Internal**	**External**	
1	MCCS-21101	Computer Fundamentals	4	0	0	**40**	**60**	**100**
2	MCCS-21102	Programming in C	4	0	0	**40**	**60**	**100**
3	MCCS-21103	Principles of Digital Electronics	4	0	0	**40**	**60**	**100**
4	MCAM-21101	Discrete Mathematics	4	0	0	**40**	**60**	**100**
5	MCCS-21104	Programming in C Laboratory	0	0	2	**60**	**40**	**100**
6	MCHU-21101	Technical Communication	3	0	0	**40**	**60**	**100**
7	MCSC-21101	Human Rights & Constitutional Duties	2	0	0	**40**	**60**	**100**
		Sub Total	**21**	**0**	**2**			
		Grand Total	**23**					**700**
		Semester II						
1	MCCS-21201	Advanced Data Structures	4	0	0	**40**	**60**	**100**
2	MCCS-21202	Advanced Computer Architecture	4	0	0	**40**	**60**	**100**
3	MCCS-21203	Advanced Operating System	4	0	0	**40**	**60**	**100**
4	MCCS-21204	Microprocessor & its Applications	4	0	0	**40**	**60**	**100**
5	MCCS-21205	Object Oriented Programming using C++	4	0	0	**40**	**60**	**100**
6	MCCS-21206	Operating System Laboratory	0	0	2	**60**	**40**	**100**
7	MCCS-21207	Object Oriented Programming using C++ Laboratory	0	0	2	**60**	**40**	**100**
		Sub Total	**20**	**0**	**4**			
		Grand Total	**24**					**700**

(Guresh Pal Singh) (Anil)

Sr. No.	Subject Code	Subject	Credits			Marks Distribution		Total Marks
		Semester III	**L**	**T**	**P**	**Internal**	**External**	
1	MCCS-21301	Advanced Software Engineering	4	0	0	**40**	**60**	**100**
2	MCCS-21302	Advanced Database Management Systems	4	0	0	**40**	**60**	**100**
3	MCCS-21303	Data Communication &Networks	4	0	0	**40**	**60**	**100**
4	MCCS-21304	Electronic Commerce	4	0	0	**40**	**60**	**100**
5	MCCS-21XXX	**Elective-I**	4	0	0	**40**	**60**	**100**
6	MCCS-21305	Advanced DBMS Laboratory	0	0	1	**60**	**40**	**100**
7	MCCS-21306	Data Communication Laboratory	0	0	2	**60**	**40**	**100**
		List of Electives-I						
	MCCS-21711	Cyber Laws and IPR	4	0	0	**40**	**60**	**100**
	MCCS-21712	Artificial Intelligence	4	0	0	**40**	**60**	**100**
	MCCS-21713	Computer Graphics	4	0	0	**40**	**60**	**100**
		Sub Total	**20**	**0**	**3**			
		Grand Total	**23**					**700**

(Guresh Pal Singh) (Anil)

Sr. No.	Subject Code	Subject	Credits			Marks Distribution		Total Marks
		Semester–IV	**L**	**T**	**P**	**Internal**	**External**	
1	MCCS-21401	Theory of Computation	4	0	0	**40**	**60**	**100**
2	MCCS-21402	Computer Based Optimization Techniques	4	0	0	**40**	**60**	**100**
3	MCCS-21403	Programming in Java	4	0	0	**40**	**60**	**100**
4	MCCS-21404	Web Technologies	4	0	0	**40**	**60**	**100**
5	MCCS-21XXX	**Elective-II**	4	0	0	**40**	**60**	**100**
6	MCCS-21405	Programming in Java Laboratory	0	0	2	**60**	**40**	**100**
7	MCCS-21406	Web Technologies Laboratory	0	0	2	**60**	**40**	**100**
		List of Electives-II						
	MCCS-21721	Cryptography and Network Security	4	0	0	**40**	**60**	**100**
	MCCS-21722	Machine Learning	4	0	0	**40**	**60**	**100**
	MCCS-21723	Digital Image Processing	4	0	0	**40**	**60**	**100**
		Sub Total	**20**	**0**	**4**			
		Grand Total	**24**					**700**
		Semester–V						
1	MCCS-21501	Programming using Python	4	0	0	**40**	**60**	**100**
2	MCCS-21502	Data Mining & Analytics	4	0	0	**40**	**60**	**100**
3	MCCS-21503	Distributed Systems	4	0	0	**40**	**60**	**100**
4	MCCS-21504	Design and Analysis of Algorithms	4	0	0	**40**	**60**	**100**
5	MCCS-21XXX	Elective-III	4	0	0	**40**	**60**	**100**
6	MCCS-21505	Programming using Python Laboratory	0	0	2	**40**	**60**	**100**
7	MCCS-21506	Seminar						**100**
		List of Electives-III						
	MCCS-21731	Deep Learning	4	0	0	**40**	**60**	**100**
	MCCS-21732	Natural Language Processing	4	0	0	**40**	**60**	**100**
	MCCS-21733	Internet of Things	4	0	0	**40**	**60**	**100**
		Sub Total	**20**	**0**	**2**			
		Grand Total		**22**				**700**
		Semester–VI						
	MCCS-21601	Major Project	0	0	24			**500**
		Sub Total	0	0	24			
		Grand Total	**24**					**500**


(Guresh Pal Singh) (Anil)

Ist

Semester

MCCS-211 01 Computer Fundamentals

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objective: The subject aims to provide the student with an understanding of basic concepts of computer science and engineering and introduction to the fundamentals of hardware, software and programming. An understanding to various emerging Technologies such as IoT, Cloud computing and Big Data.

1. Introduction to Computer: Definition, Computer Hardware & Computer Software Components: Hardware – Introduction, Input devices, Output devices, Central Processing Unit, Memory- Primary and Secondary. Software - Introduction, Types – System and Application. (4)

2. Programming Paradigms and Development Tools: – Problem Analysis, Program Constructs Sequential, Decision, Loop), Algorithms, Flowcharts, Pseudocode. Decision table, Modular Programming, Top– down and Bottom–up Approaches, functional, Procedural object–oriented, and logic programming, Programming Languages – Syntax & Semantics. (8)

3. Operating system: Definition, Functions, Types, Classification, Elements of command based and GUI based operating system. Computer Network: Overview, Types (LAN, WAN and MAN), Data communication, topologies. (8)

4. Internet: Overview, Architecture, Functioning, Basic services like WWW, FTP, Telnet, Gopher etc., Search engines, E-mail, Web Browsers. (4)

5. Internet of Things (IoT): Definition, Sensors, their types and features, Smart Cities, Industrial Internet of Things. (4)

6. Emerging Technologies: Applications and use cases Cloud Computing: Nature and benefits, AWS, Google, Microsoft & IBM Services, Virtual Reality, Grid computing, Green computing, Big data analytics, Quantum Computing and Brain Computer Interface. (8)


Course Outcomes:

After completing this course student will be able to:

1. Demonstrate the knowledge of the basic structure, components, features and generations of computers.
2. Describe the concept of computer languages, language translators and construct algorithms
3. Compare and contrast features, functioning & types of operating system and computer networks.
4. Demonstrate architecture, functioning & services of the Internet. Illustrate the emerging trends and technologies in the field of Information Technology.

Suggested Books:

1. Computers Today: Suresh K. Basandra, Galgotia, Updated Edition,2012.
2. Gurvinder Singh & Rachhpal Singh: A Test Book on Windows Based Computer Courses, Kalyani Publishers, 10th Edition 2008.
3. Norton, Peter: Introduction to Computers, McGraw Hill
4. Martin, James: Telecommunications and the Computer, PHI
5. Distributed and Cloud Computing, 1st edition, Morgan Kaufmann, 2011 by Hwang & Dongarra & Fox

MCCS-21102 Programming in C

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objective:

1. To learn the fundamental programming concepts and methodologies which are essential to building good C programs.
2. To practice the fundamental programming methodologies in the C programming language via laboratory experiences.
3. To code, document, test, and implement a well-structured, robust computer program using the C programming language.
4. To write reusable modules (collections of functions).

1. Introduction to Program Development: Problem Analysis, Designing a solution**.**

Overview of C: Brief history of C, introduction to different versions of C. General Structure of a C program, stages in the development of a C program**.**


(Guresh Pal Singh) (Anil)

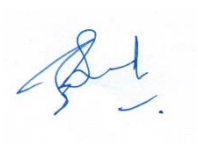
Data Types, Operators & Expressions: Constants and variables, data types, declaring variables, storage classes, different types of expressions and their evaluation, conditional expression, assignment statement, enumerated data type, redefining/creating data types, type casting.
Console Input/Output: Standard input/output devices, unformatted input/output functions (character I/O functions and string I/O functions), formatted input/output functions (scanf( ) function and printf ( ) function). (6)

2. Control Statements: Decision making using if, if – else, elseif and switch statements, Looping using for, while and do – while statements, transferring program control using break and continue statements
Arrays & Strings: Introduction to arrays, declaring arrays, initializing arrays, processing of arrays, introduction to strings. Structures & Unions: Introduction to structures, declaring structures, initializing structures, accessing elements of structures, array of structures, nested structures, passing structures as arguments to a function, introduction to unions**.** (10)

3. Functions: Defining a function, local variables, return statement, invoking a function, specifying and passing arguments (including arrays, strings) to a function, function prototyping and use of header files, building own library, recursion.
Pointers: Why pointers? Declaring pointers, accessing values via pointers, pointer arithmetic, pointers to arrays, Array of pointers, pointers to strings, pointers to structures, self–referential structures.
Program Structure: Program structure revisited, managing multi–file programs using traditional approach of separate compilations and project facility of Turbo C compiler, storage classes revisited.
File I/O: Introduction to files, different ways of file processing (standard I/O & system I/O), description of various library functions for file handling, updating files. (10)

4. Introduction to Object Oriented Paradigm – Object Oriented programming and C++, Structured Programming methodology, its shortcomings, advantages of OOPS (Object Oriented Programming Style). OOP concepts: Abstraction, Encapsulation, Data Hiding, Inheritance, Polymorphism. Overview of C++ – Data types, Input/output statements, Flow of control – looping statements, branching statements, Pointers & references, namespaces. (10)

5. Class design: constructors, destructors, operator overloading, reuse through inheritance, virtual functions, exception handling. I/O with stream classes, memory management
The Standard Template Library (STL): containers, algorithms, iterators, adaptors, function objects (4)




(Guresh Pal Singh) (Anil)

Course Outcomes:

After completing this course student will be able to:

1. Describe the advantages of a high level language like C, the programming process, and the compilation process**.**
2. To describe and use software tools in the programming process**.**
3. To apply good programming principles to the design and implementation of C programs**.**
4. To design, implement, debug and test programs using the fundamental elements of C
5. To demonstrate an understanding of primitive data types, values, operators and expressions in C

Suggested Books:

1. E.Balagurusamy "Programming in C". Tata McGraw Hill
2. Y. Kanetkar "Let Us C". BPB publication
3. Ashok N. Kamthane "Programming with ANSI and TURBO C". Pearson Education
4. Lafore R, Object Oriented Programming, Third Edition, Galgotia Publications
5. Byron S. Gottfried, Programming in C, Second Edition, McGraw Hills.
6. R.S. Salaria,Problem Solving and Programming in C, Second Edition

MCCS-21103 Principal of Digital Electronics

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objective:

1. To present the Digital fundamentals, Boolean algebra and its applications in digital systems
2. To familiarize with the design of various combinational digital circuits using logic gates
3. To introduce the analysis and design procedures for synchronous and asynchronous sequential circuits
4. To explain the various semiconductor memories and related technology

1. Introduction

Junction diodes, Bipolar Transistor and biasing techniques, FET, number systems-binary, signed binary, octal hexadecimal number, binary arithmetic, one's and two's complements arithmetic, codes, error detecting and correcting codes, logic gates, Demorgans theorem, universal gates. (8)

2. Digital Circuit I

Standard representation for logic functions, K-map representation and simplification of logic functions using K-map, minimization of logical functions. Don't care conditions, Multiplexer, De-Multiplexer, Encoders, Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder, array multiplier. (12)


(Guresh Pal Singh) (Anil)

3. Digital Circuit II

SR flip flop, J- K-T and D types flip flops, applications of flip flops, shift registers, synchronous counters, Asynchronous counters, A/D and D/A converter (10)

4. Digital Memories

classification and characteristics of memories, sequential memory, read only memory (ROM), read and write memory(RAM), content addressable memory (CAM), charge de coupled device memory (CCD), commonly used memory chips, ROM as a PLD, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA). (8)

Course Outcome:

After completing this course student will be able to:

1. Describe the number systems, conversions and their applications.
2. Apply minimization techniques such as K maps, Tabular method etc for the design of digital circuits.
3. Design combinational and sequential circuits.
4. Differentiate various type of memories and there use in different applications.

Suggested Books:

1. R. P. Jain, "Modern Digital Electronics", McGraw Hill Education, 2009.
2. M. M. Mano, "Digital logic and Computer design", Pearson Education India, 2016.

Reference Books

1. A. Kumar, "Fundamentals of Digital Circuits", Prentice Hall India, 2016.

MCAM-21101 Discrete Mathematics

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objectives: To provide knowledge of combinatorial problems, algebraic structures and graph theory required for building mathematical foundation of computer science.

1. Logic and Propositional Calculus: Proposition and Compound Propositions, basic Logical Operations, Propositions and Truth Tables, Tautologies and Contradictions, Logical Equivalence, Duality law, Algebra of propositions, Conditional and Biconditional Statements, Arguments, Logical Implication, Propositional Functions, Predicates and Quantifiers, Negation of Quantified Statements, Inference theory of the predicates calculus.

Algebraic Systems: Operations, Semigroups, Groups, Subgroups, Normal Subgroups and Homomorphisms, Rings, Fields, Polynomials over a field. (8)

2. Graph Theory: Graphs and Multigraphs, Subgraphs, Isomorphic and Homeomorphic Graphs, Walks, Paths and Cycles, Eulerian Graph, Hamiltonian Graph, Connectivity, Bridges of Konigsberg, Traversable Multigraphs, Labeled and Weighted Graphs, Complete, regular and Bipartite Graphs, Tree graphs, Planar Graphs, Regions, Euler's Formula, Graph Colorings, Chromatic Number, Welch–Powell Algorithm, Representing Graphs in Computer Memory. (10)

3. Recurrence Relations and Generating Functions: Polynomial expressions, recursion theorem, closed form expression, generating function, solution of recurrence relation using generating function. (8)

4. Properties of the Integers: Order and Inequalities, Absolute Value, Mathematical Induction, Division Algorithm, Greatest Common Divisor, Euclidean Algorithm, Fundamental Theorem of Arithmetic, Congruence Relation, Congruence Equations. (8)

5. Boolean Algebra: Boolean algebra and its duality, Duality, Boolean Algebra as Lattices, Boolean identities, sub–algebra, Representation Theorem, Sum–of–Products Form for Sets, Sum–of Products Form for Boolean Algebra, Minimal Boolean Expressions, Prime Implicants, Logic Gates and Circuits, Boolean Functions, Karnaugh Maps. (8)


(Guresh Pal Singh) (Anil)

Course Outcomes

After completing this course student will be able to:

1. Write an argument using logical notation and determine if the argument is or is not valid.

2. Understand the basic principles of sets and operations in sets..

3. Demonstrate an understanding of relations and functions and be able to determine their properties.

4. Demonstrate different traversal methods for trees and graphs.

5. Model problems in Computer Science using graphs and trees.

Text Books:

1. Trambley,J.P.andManohar,R:DiscreteMathematicalStructureswithApplicationstoComputerScience,TataMcGraw–HillEducation(2001).

2. LiuC.L.:ElementsofDiscreteMathematics,McgrawHill(2008).

3. AlanDoerrandKennethLevasseur:AppliedDiscreteStructuresVersion2.0,UniversityofMassachusettsLowell(2013).

4. Narsingh Deo: Graph Theory with Applications to Engineering and Computer Science, Prentice –Hall of India(2007).

5. Lipschutz, S.and Lipson,M.: Discrete Mathematics, Schaum's outlines series

6. (Revised Third Edition),McGraw Hill Education(2010).14

(Guresh Pal Singh) (Anil)

MCCS-21104 Programming in C

Internal Marks: 60 **L T P**
External Marks: 40 **0 0 4**
Total Marks: 100

1. Write C program to input and output the text message.

2. Write C Program to perform all arithmetic operations.

3. Write C Program to utilize the math function.

4. Write C Program to perform the mathematical expressions.

5. Write C Program for Local and Global Variables.

6. Write C Program for internal static and external static variables.

7. Write C Program to find the roots of a Quadratic equation.

8. Write C Programs for all the Operators. (Arithmetical, Logical, Relational, Bitwise).

9. Write C Programs for Increment and Decrement Operators.

10. Write C Programs to implement the Ternary Operator.

11. Write C Programs for special Operators.

12. Write C Programs for all the Control Structures. (Sequential Control Structures, Conditional Control Structures, Iterative Control Structures).

13. Write C Programs to display the different types of patterns using nested for loop.

14. Write C Program for Statements. (switch, break, goto, continue etc.,).

15. Write C Program to print biggest number from n numbers.

16. Write a C Program to find the given integer number is even or odd number.

17. Write a C Program to calculate the factorial of a given number.


(Guresh Pal Singh) (Anil)

18. Write a C Program to swap the two numbers using temp variable and without using temp variable.

19. Reading and Printing a single dimensional array of elements.

20. Ascending and descending of an array.

21. Sum of all odd numbers and sum of all even numbers in a single dimensional array.

22. Mathematical operations on single dimensional arrays.

23. Reading and Printing a multi dimensional array of elements.

24. Mathematical operations on multi dimensional array of elements.

25. Passing an array element to a function.

26. Reading and Printing a string.

27. C Programs on String functions.

28. Write a C program to calculate string length by writing the user-define function.

29. Function declaration and initialization.

30. C Program to differentiate the parameters and arguments in functions.

31. Programs for different types of inbuilt functions.32. Call by value and Call by reference programs in functions.

33. Write a program to swap the given 2 number using passing by reference.

34. Write C Programs to perform all valid arithmetic operations using pointers.

35. C programs on Structures and accessing of members of the structures.

36. Write a C program to print a book information (Book name, Book no, author name) by writing a structure.

37. Write a C program by passing structure elements to a function and display employee information (emp no, emp name, emp salary, and emp address).

38. C Programs on Reading a file from the secondary storage device.

39. C Program on writing and appending a file on the secondary storage device.

40. C Program on Opening and closing a file.

Text Books:

1. C programming and Data Structures, P. Padmanabham, Third Edition, BS Publications.

2. Let Us C by Yashwanth Kanethar.

3. "Programming in ANSI C" by E. Balaguruswamy.

4. Programming in C, 2nd Edition, Oxford by Pradip Dey, Mannas Ghosh.

MCHU-101 Technical Communication

Internal Marks: 40 **L T P**
External Marks: 60 **3 0 0**
Total Marks: 100

Course Objective
The goal of this course is to prepare engineering students with the individual and collaborative technical writing, presentation, and research skills necessary to be effective technical communicators in academic and professional environments.

1. Information Design and Development
Different kinds of technical documents, Information development life cycle, Organization structures, factors affecting information and document design, Strategies for organization, Information design and writing for print and for online media. (7)

2. Technical Writing, Grammar and Editing
Technical writing process, forms of discourse, Writing drafts and revising, Collaborative writing, creating indexes, technical writing style and language. Basics of grammar, study of advanced grammar, editing strategies to achieve appropriate technical style. Introduction to advanced technical communication, Usability, Hunan factors, Managing technical communication projects, time estimation, Single sourcing, Localization. (7)

3. Self Development and Assessment
Self assessment, Awareness, Perception and Attitudes, Values and belief, Personal goal setting, career planning, Self-esteem. Managing Time; Personal memory, Rapid reading, Taking notes; Complex problem solving; Creativity.
(7)


4. Communication and Technical Writing

Public speaking, Group discussion, Oral; presentation, Interviews, Graphic presentation, Presentation aids, Personality Development. Writing reports, project proposals, brochures, newsletters, technical articles, manuals, official notes, business letters, memos, progress reports, minutes of meetings, event report. (7)

5. Ethics

Business ethics, Etiquettes in social and office settings, Email etiquettes, Telephone Etiquettes, Engineering ethics, Managing time, Role and responsibility of engineer, Work culture in jobs, Personal memory, Rapid reading, Taking notes, Complex problem solving, Creativity. (7)

Course Outcomes

At the conclusion of this course, students will demonstrate proficiency by:

1. Understanding the characteristics of technical writing and the importance of purpose, audience, and genre for written communication in technical fields.
2. Articulating complex engineering ideas appropriate for targeted audiences.
3. Planning, drafting, revising, editing, and critiquing technical and professional documents through individual and collaborative writing.
4. Writing effective technical and business documents that are grammatically and stylistically correct.
5. Preparing and delivering professional technical presentations through applying principles of effective oral communication and slide design.
6. Applying principles for the visual display of quantitative information.

Books Recommended-

1. David F. Beer and David McMurrey, Guide to writing as an Engineer, John Willey. New York, 2004
2. Shiv Khera, You Can Win, Macmillan Books, New York, 2003.
3. Raman Sharma, Technical Communications, Oxford Publication, London, 2004.
4. Sharma, R. and Mohan, K. Business Correspondence and Report Writing, TMH New Delhi 2002.

MCSC-21101 Human Rights & Constitutional Duties

Internal Marks: 40 **L T P**
External Marks: 60 **2 0 0**
Total Marks: 100

Course Objective:
In-depth insight into the constitutional, statutory and institutional aspects of human rights protection in India. Covers constitutional provisions dealing with human rights and special legislations dealing with protection of vulnerable and marginalized groups.

1. Introduction to Human Rights

Foundational Aspects: Meaning, Nature, Classification

Evolution of the Concept: Magna Carta to Universal Declaration of Human Rights; Generations of Human Rights (6)

2. Introduction to Human Duties

Conceptual Perspective: Meaning, Nature & Characteristics of Human Duties; Classification of Human Duties; Relevance of Human Duties

Human Duties in India: Fundamental Duties in Indian Constitution Part IV A

(a) to abide by the Constitution and respect its ideals and institutions, the National Flag and the National Anthem;

(b) to cherish and follow the noble ideals which inspired our national struggle for freedom;

(c) to uphold and protect the sovereignty, unity and integrity of India;

(d) to defend the country and render national service when called upon to do so;

(e) to promote harmony and the spirit of common brotherhood amongst all the people of India transcending religious, linguistic and regional or sectional diversities; to renounce practices derogatory to the dignity of women;

(f) to value and preserve the rich heritage of our composite culture;

(g) to protect and improve the natural environment including forests, lakes, rivers and wildlife, and to have compassion for living creatures;

(h) to develop the scientific temper, humanism and the spirit of inquiry and reform;

(i) to safeguard public property and to abjure violence;

(j) to strive towards excellence in all spheres of individual and collective activity so that the nation constantly rises to higher levels of endeavour and achievement;

(k) who is a parent or guardian to provide opportunities for education to his child or, as thecase may be, ward between the age of six and fourteen years.) (10)

3. Concept of Human Rights in India

Constitutional-Legal Framework: Fundamental Rights; Directive Principles of State Policy Governmental Institutions for the Protection of Human Rights: Working of National HumanRights Commission; National Commission for Women. (8)

4. Actual Status of Human Rights in India

Status of Economic Social & Cultural Rights in India: Violence against Women; Violation of Child Rights: An Appraisal.

State of Civil & Political Rights in India: A study of Jammu & Kashmir and the North-East. (8)

Course Outcomes:

After completing this course student will be able to:

1. Demonstrate a good understanding of the provisions under the Constitution of India dealing with human rights.
2. Display a good understanding of the nature and scope of special legislations dealing with protection of human rights of marginalized and vulnerable sections.
3. Demonstrate a good understanding of the practical application of human rights law to specific human rights problems in India.
4. Analyze complex human rights problems and apply relevant provisions of human rights law in India

Readings List

1. United Nations. The *United Nations and Human Rights 1945-1995*. Geneva: UnitedNations Blue Books Series, Vol. VII, 1996.
2. Sastry, S. N. *Introduction to Human Rights and Duties*. Pune: University of Pune Press,2011.
3. Mertus, Julie. *The United Nations and Human Rights-A Guide for a New Era*. London: Routledge, 2009.
4. Donnelly, Jack. *Universal Human Rights in Theory and Practice*. New York: Cornell University Press, 2013.
5. Hammarberg, Thomas. *Taking Duties Seriously- Individual Duties in International Humanitarian Law*. Versoix: International Council on Human Policy, 1999.
6. Miller P. Frederic, *et al. Fundamental Rights, Directive Principles and FundamentalDuties in India*. New York: VDM Publishing, 2009.
7. Cinganelli, Davis Louis. *Human Rights- Theory and Measurements*. London: MacmillanPress, 1988.
8. Ishay, M. R. *The History of Human Rights*. New Delhi: Orient Longman, 2004.
9. Mohapatra, Arun Ray. *National Human Rights Commission of India: Formation,Functioning and Future Perspectives*. New Delhi: Atlantic, 2004.
10. Deol, Satnam Singh. *Human Rights in India-Theory and Practice*. New Delhi: Serials Publications, 2011.
11. Nessa, Saifun, et al. *Human Rights- With Special Reference to North East*. New Delhi: Reliance Publishing House, 2007.
12. Haragopal G. & B. Jagannatham. "Terrorism and Human Rights- The Indian Experiencewith Repressive Laws". *Economic and Political Weekly*, Vol. 44, No. 28, 2009.
13. JKHRDC (J & K Human Rights Awareness and Documentation Centre). *Human Rights Abuses in Kashmir (a collection of case study reports of human rights violation in Kashmir)*. Sri Nagar: IKS Publishers & Distributers, 2000.
14. Human Rights Watch. *With Friends like these: Human Rights Violations in Kashmir*. Washington: Human Rights Watch Publication, 2006.
15. Asia Watch & Physicians for Human Rights. *Human Rights Crisis in Kashmir*. Washington: Human Rights Watch Publication, 1993.
16. Chadrashekhar, Mamta, *Human Rights, Women & Violation*. New Delhi: Educreation Publishing, 2016.
17. Chopra, Geeta. Child Rights in India- Challenges and Social Action. New Delhi: Springer, 2015.


(Guresh Pal Singh) (Anil)

2nd Semester

MCCS-21201 Advanced Data Structures

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objective: This course is designed to understand the basic and advanced concepts of data structures and to highlight the importance of data structures in developing and implementing efficient algorithms. Another objective of the course is to develop ability in students to design algorithms for real life problems.

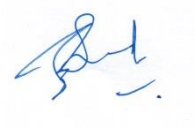
1. Introduction to Data Structures
Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Linear Search and Binary Search Techniques (4)

2. Types of Data Structures
Introduction to Concept and Operations: Stacks, Queues, Linked Lists
Graph: Basic Terminologies and Representations, Graph traversal algorithms and complexity analysis.
Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree,
Balanced trees: red-black trees, B-trees, AVL Trees, 2-3 trees, 2-3-4 trees. (8)

3. Sorting, Hashing & String Matching
Sorting and Hashing: Objective and properties of different sorting algorithms: Bubble Sort, Quick Sort, Hashing. Amortized analysis, string matching. (8)

4. Heaps
Binary heaps, heap operations, implementation and applications.
Binomial Heaps, Fibonacci Heaps (6)

5. External Data Structures
External data structures - external storage, external files, external sorting searching indexing files, external hashing. (4)




Course Outcomes:

After completing this course student will be able to:

1. Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms.
2. Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs
3. Compare alternative implementations of data structures with respect to performance
4. Design and implement an appropriate hashing function for an application
5. Discuss the computational efficiency of the principal algorithms for sorting, searching, and hash

Suggested Books:

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
2. Introduction to algorithms: Cormen, Leiserson, Rivest and Stein

MCCS-21202 Advance Computer Architecture

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objective:

1. To impart basic concepts of computer architecture and organization,
2. To explain key skills of constructing cost-effective computer systems.
3. To familiarize the basic CPU organization.
4. To help students in understanding various memory devices.

1. Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logic shift unit, Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit. (8)

2. Computer Instruction Set; Instruction codes, Computer Registers, Computer Instructions and Instruction cycle. Timing and Control, Memory-Reference Instructions, Input-Output and interrupt, Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Complex Instruction Set Computer (CISC) Reduced Instruction Set Computer (RISC), CISC vs RISC (9)

3. Memory And Input Output Techniques: I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA, RAM,ROM, associative memory, cache memory and mapping techniques, virtual memory. (8)

4. Parallel Processing: The Use of Multiple Processors, Symmetric Multiprocessors, Cache Coherence and the MESI Protocol, Multithreading and Chip Multiprocessors, Clusters, Nonuniform Memory Access Computers. (9)

Course Outcome:

After completing this course student will be able to:

1. Identify various components of computer and their interconnection

2. Identify basic components and design of the CPU: the ALU and control unit.

3. Compare and select various Memory devices as per requirement.

4. Compare various types of IO mapping techniques

Suggested Books:

1. M. Moris Mano (2006), Computer System Architecture, 3rd edition, Pearson/PHI, India.
2.William Stallings (2010), Computer Organization and Architecture- designing for performance, 8th edition, Prentice Hall, New Jersy

Reference Books

1. Carl Hamacher, Zvonks Vranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India.
2. John P. Hayes (1998), Computer Architecture and Organization, 3rd edition, Tata McGrawHill.

MCCS-21203 Advanced Operating Systems

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objectives:

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To know the components and management aspects of concurrency management
6. To learn to implement simple OS mechanisms

1. Introduction

Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System. (4)

2. Processes

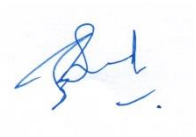
Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads, Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Preemptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF. (6)

3. Inter-process Communication

Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc. (6)

4. Deadlocks

Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery. (6)




5. Memory Management

Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging. Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU). (6)

6. I/O Hardware

I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks

(6)

7. Multi-Processor and Distributed Operating System

Introduction, Architecture, Organization, Resource sharing, Load Balancing, Availability and Fault Tolerance, Design and Development Challenges, Inter-process Communication, Distributed Applications – Logical Clock, Mutual Exclusion, Distributed File System, Real Time Operating System. (4)

Course Outcomes:

After completing this course student will be able to:

1. Demonstrate understanding of design issues of advanced operating systems and compare different types of operating systems.
2. Analyze design aspects and data structures used for different subsystems of Unix OS.
3. Demonstrate understanding of different architectures used in Distributed OS.
4. Demonstrate understanding of different architectures used in Multiprocessor OS and analyze the design and data structures and various processor scheduling algorithms.
5. Classify Real Time OS and analyze various real time scheduling algorithms.

Suggested books:

1. Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Suggested reference books:

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison- Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

MCCS-21204 Microprocessor & Its Applications

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objectives

1. To demonstrate how fundamental high-level programming constructs are implemented at the machine-language level.
2. To write assembly language program that can input, process and output results.
3. To demonstrate the basic computer architecture.
4. Students will be to apply mathematical foundations, algorithmic principles, and computer science theory to the modeling and design of computer based systems

1. Introduction: Overview of Microcomputer Systems: Hardware – Software, Addresses – General Operation of a Computer – Microprocessors in Digital System Design. 8086 Architecture: CPU Architecture – Internal Operation, Machine Language Instruction: Addressing modes – Instruction Formats. (8)

2. 8086 Instruction Set and Programming: 8086 instruction set and description, assembler directives, simple sequence programs, jump, flag and conditional jump, if-then, if-then-else,multiple if-then-else programs, while do programs. (9)

3. Interface Design: Serial Communication Interfaces: Asynchronous Communication – Synchronous Communication – Physical Communication Standards – 8251A Programmable Communication Interface, Parallel Communication: 8255A Programmable Peripheral Interface - A/D and D/A Example, Programmable Timers and Event Counters: Intel's 8254 Programmable Interval Timer – Interval Timer Application to A/D, DMA Controllers. (8)

4. Advanced Microprocessors: The 80386 : Introduction – Operating Modes – Processor Model – Programming Model, The 80486 : Introduction – Processor Model – Programming Model, The Pentium : Introduction – Processor Model – Programming Model (9)

Course Outcome:

After completing this course student will be able to:

1. Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level.
2. Write assembly language program that can input, process and output results.
3. Demonstrate the basic computer architecture.
4. Students will be to apply mathematical foundations, algorithmic principles, and computer science theory to the modeling and design of computer based systems

Suggested Books:

1. Yu-Cheng Liu, Glenn A Gibson, "Microcomputer Systems: The 8086/8088 Family", Second Edition, Pearson Education
2. MDouglas V Hall, "Microprocessors and Interfacing", Second Edition, TMH

Reference Books

1. Kenneth J. Ayala, "The 8086 Microprocessor : Programming & Interfacing The PC", Cengage Learning (2008)
2. Barry B. Brey, "The Intel Microprocessors", Seventh Edition, PHI


(Guresh Pal Singh) (Anil)

MCCS-21205 Object Oriented Programming using C++

Internal Marks: 40 **L T P**
External Marks: 60 **4 0 0**
Total Marks: 100

Course Objective: To understand the basic concepts of object oriented programming languages and to learn the techniques of software development in C++.

Detailed contents:

1. Object-Oriented Programming Concepts (2)
Introduction, Comparison between procedural programming paradigm and object-oriented programming paradigm, Basic concepts of object oriented programming

2. Classes and Objects (4)
Classes and Objects: Specifying a class, creating class objects, accessing class members, access specifiers, static members, friends of a class,

Constructors and Destructors: Types of Constructors, Constructor Overloading, Destructors

3. Function & Operator Overloading (4)
Function Overloading, Default Arguments, Ambiguity in Function Overloading.

Operator Overloading: Overloading unary and binary operators, Type Conversion using Operator Overloading

4. Inheritance (6)
Concept of inheritance, Base & derived classes, Access Specifiers, Class Hierarchies, Types of Inheritance with examples.

5. Virtual Functions and Polymorphism (4)
Virtual functions, friend functions, static function, this pointer, polymorphism, Types of Polymorphism with examples, templates, class templates.

6. Exception Handling & Generic Programming (6)
Review of traditional error handling, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions.

Templates and Generic Programming: Template concepts, Function templates, class templates, illustrative examples.


(Guresh Pal Singh) (Anil)

7. File Handling & GUI (4)

File streams, hierarchy of file stream classes, error handling during file operations, Reading/writing of files, accessing records randomly, updating files.

Working with GUI

Course Outcomes:

After completing this course student will be able to:

1. Inculcate the principles of object-oriented problem solving and programming.
2. Outline the essential features and elements of the C++ programming language.
3. Explain programming fundamentals, including statement and control flow and recursion.
4. Apply the concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism.
5. Analyze problems and implement simple C++ applications using an object-oriented software engineering approach.

Suggested Text Books:

1. Lafore R., Object Oriented Programming in C++, Waite Group.
2. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill.

Suggested Reference Books

1. R. S. Salaria,Mastering Object-Oriented Programming with C++, SalariaPublishingHouse.
2. BjarneStroustrup, The C++ Programming Language, AddisonWesley.
3. Herbert Schildt, The Complete Reference to C++ Language,McGraw Hill-Osborne.
4. Lippman F. B, C++ Primer, AddisonWesley

MCCS-21206 Advanced Operating Systems Laboratory

Internal Marks: 60 **L T P**
External Marks: 40 **0 0 4**
Total Marks: 100

1. To explore and installation process different operating systems like Linux, Windows etc.
2. Virtualization, Installation of Virtual Machine Software and installation of Operating System on Virtual Machine
3. Execute various basic and advance Linux commands, commands for files and directories, creating and viewing files, File comparisons, Disk related commands.
4. Execute Linux commands for Processes in Linux, connecting processes with pipes, background processes, managing multiple processes.
5. Study and usage of vi Editor.
6. Basics of Shell programming, various types of shell, Shell Programming in bash.
7. Study and implementation of shell variables, shell keywords.
8. Implement conditional statements, looping statement and case statement in Shell programming.
9. Implement parameter passing and arguments in Shell programming.
10. Implement Shell programs for automate system tasks and report printing.

MCCS-21207 Object Oriented Programming using C++ Laboratory

Internal Marks: 60 **L T P**
External Marks: 40 **0 0 4**
Total Marks: 100

1. Write programs to understand the basic of object oriented programming.
2. Write program to demonstrate the use of constructors and destructors
3. Write a program to demonstrate the overloading of increment and decrement operators.
4. Write a program to demonstrate the overloading of binary operators.
5. Write a program to demonstrate the overloading of increment and decrement operators using friend function.
6. Write a program to demonstrate the typecasting of class type to class type & basic type.
7. Write a program to demonstrate the multilevel inheritance.
8. Write a program to demonstrate the multiple inheritances.


9. Write a program to demonstrate the runtime polymorphism.
10. Write a program to demonstrate the exception handling.
11. Write a program to demonstrate the reading and writing of mixed type of data.
12. Write a program to demonstrate the reading and writing of objects.
13. Write a menu driven program that implement following operations (using separate functions) on a linear array:
14. Insert a new element at end as well as at a given position
15. Delete an element from a given whose value is given or whose position is given
16. To find the location of a given element
17. To display the elements of the linear array
18. Write a program to implement various operations on stack
19. Write a program to demonstrate the implementation of various operations on a linear queue represented using a linear array.
20. Write a program to demonstrate the implementation of various operations on a circular queue represented using a linear array.
21. Write a menu driven program that maintains a linear linked list whose elements are stored in on ascending order and implements the following operations (using separate functions):
a. Insert a new element
b. Delete an existing element
c. Search an element
d. Display all the elements
22. Write a program to implement binary search technique.