CURRICULA-2020

MCA

(2 Years - CBCS Scheme)

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OFNATURAL SCIENCES

Jamia Millia Islamia: New Delhi, India

Recommended by the Board of studies DOS-FNS-JMI on 26/08/2020

MCA (2 Years): 2020

A.	**Reference**	1. Notification No.AICTE/AB/MCA/2020-21/03-07020) 2. Adaptation by BOS dated 06-07-20 for MCA(2-Years) program.
B.	**Motivation**	1. Rapidly changing academic and technological scenarios around the world. 2. Conformance to prescriptions from the regulatory bodies, and JMI ordinances. 3. Keeping abreast with the current and future industrial skill-set requirements, and placement related tests.
C.	**Bases and Constraints**	UGC Guideline on Choice Based Credit Based System (CBCS) – with greater autonomy to students on the selection of courses – by reinforcing the following: 1. Retaining prescribed types as Core, Electives, CBCS, Ability/Skill enhancement courses. 2. Retaining the 4thsemeater as full project-semester, because of the mandate of MCA as a professional program and placements. 3. CBCS courses are to be open for any PG students including DCS students, subject to availability of seats. 4. One CBCS-Elective and CBCS-Ability/Skill-Enhancement course each, to be compulsory in each of the semesters (1-3).
D.	**Abbreviations**	Computer Science: **CS**, Core: C, Elective: E, Ability/Skill: S
E.	**Course Codes**	Unique code to be assigned to each of the typical courses at (UG/PG level separately), offered by the Department using the following coding scheme: • Computer Science Core/elective: CSC/E • Computer Science (CBCS Elective): CBCSE • Computer Science (CBCS Ability/Skill-Enhancement):CBCSS
F.	**Course L-T-P**	1. Core Theory Courses: 3-0-0 2. Lab-Oriented Theory Courses (Electives): 3-0-2 3. Lab-Courses: 0-0-4 4. CBCS Elective Courses: 4-0-0/3-0-2 5. Ability/Skill Enhancement Courses: 2-0-2 6. Major Project: 0-4-32
G.	**Eligibility for Admission**	1. Bachelor's Degree with at least 50% Marks in (a) Computer Science/Engineering/Applications/Equivalent Allied, OR (b) any other discipline with Mathematics at 10+2 Level. 2. In case of 1(b), a candidate must either produce 'a valid certificate of passing at least 6 credits of Computer Science courses (with at least C grade/50% marks) from any Govt-approved mode at 10+2/graduation level' **OR** 'complete the bridge courses from the Department in the first semester'.
H.	**Bridge Courses**	Applicable to those qualifying under G1(b), under which a student must produce or complete at least 6 credits of the courses as advised by the department, necessarily including a course on programming.
I.	**Special Considerations**	• Accommodation of non-Computer Science graduates and the bridge-courses. • Applied discipline and professional nature of programs. • Balancing academic, technological, and industrial imperatives. • National and global connect. • ACM, UGC, AICTE, and other central universities' curriculum.
J.	**Track Specialization**	MCA degree may be awarded in a specialization (Advanced Computing OR Informatics), provided a student, besides fulfilling all the requirements: (i) opts all the electives from that track, and (ii) requests formally in writing to the Department.
K.	**Remarks**	1. Students are encouraged to enroll in courses about Communication Skills and Management from other Departments, as CBCS electives. 2. CBCS courses of minimum 3 credits each that may be chosen from other departments subject to students' requirements and convenience. 3. Lab and project courses shall have independent practical and viva-voce examinations. 4. At least two courses shall be offered for each of the typical electives, provided at least 15 students opt for a course. 5. Faculty members may be allocated 2 to 4 lab-periods/week subject to feasibility.

		6. Department may float any other elective, beyond the listed ones, subject to feasibility and endorsement of BOS.

MCA(2 Years) Programme Structure: 2020

SEM	CODE	COURSE TITLE	CREDITS	REMARKS
Bridge Courses	CSC01	Computer Fundamentals	4	Any two courses as advised by the Department, as per clause H.
	CSC02	Programming and Problem Solving using C		
	CSC03	Applied Operating Systems	3	
	CSC04	Information Systems	3	
I	CSC11	Digital Logic and Computer Architecture	3	Periods/Week: ~24 Credits: ~20
	CSC12	Discrete Mathematics	3	
	CSC13	Algorithmics and Program Design	3	
	CSC14	Database Management Systems	3	
	CSC15	Lab-I (APD)	2	
	CSC16	Lab-II (DBMS)	2	
	CBCSE17	CBCSE-I	4	
II	CSC21	Software Engineering	3	Periods/Week: ~31 Credits:~27
	CSC22	Object Oriented Programming	3	
	CSC23	Advanced Data Structures	3	
	CSC24	Operating Systems and Shell Programming	3	
	CSE25	Elective-I	4	
	CSC26	Lab-III (OOP)	2	
	CSC27	Lab-IV(ADS+SP)	2	
	CBCSE28	CBCSE-II	4	
	CBCSS29	CbcsS-I	3	
III	CSC31	Data Communication and Networks	3	Periods/Week: ~31 Credits: ~27
	CSC32	Artificial Intelligence	3	
	CSC33	Information Security	3	
	CSC34	Analysis and Design of Algorithms	3	
	CSE35	Elective-II	4	
	CSC36	Lab-V (AI)	2	
	CSC37	Lab-VI (ADA)	2	
	CBCSE38	CBCSE-III	4	
	CBCSS39	CbcsS-II	3	
IV	CSC41	Major Project	20	Periods/Week: 36 and Credits: 20
Summary: Core-Courses(12)+Elective(2)+Lab-Courses(6)+CBCS-Elective(3)+CBCS-Skill(2)+Major-Project (1)=26 Courses				

Bi-Track Specialization: Optional Model

Department Elective Courses for PG Students: L-T-P: 4-0-0 /3-0-2 (4 Credits)				
Specialization Track		**Advanced Computing**	**Informatics**	
II	CSE25	**Advanced DBMS** **Computer Graphics** **Theory of Computation**	**Software Quality Assurance** ~~Decision Support Systems~~ ~~Software Project Management~~	~~Management Information Systems~~ **Data Mining and Warehousing** **Software Testing and Verification**
III	CSE35	**Distributed Systems** **Cryptography** ~~Compiler Design~~ **System Software** **Digital Image Processing**	~~Soft Computing~~ **Cyber Security** **Cloud Computing** **Machine Learning** ~~Security Audit and Design~~	~~Eduinformatics~~ ~~Health Informatics~~ ~~Advanced Software Engineering~~ **Adhoc Networks** **Programming with Java**
CBCS Courses for PG Students: L-T-P: 4-0-0/3-0-2 (4 Credits)				
I	CBCSE17	Statistical Computing	e-Business Systems	Human Computer Interaction
II	CBCSE28	Modelling and Simulation	Business Informatics	Social Informatics
III	CBCSE38	~~Numerical Optimization~~	~~IT Management~~	~~Multimedia Applications~~
CBCS (Ability/Skill Enhancement) Courses for PG Students: L-T-P: 2-0-2 (3 Credits)				
II	CBCSS29	Programming with Python	Mobile Applications	~~Portal Development~~
III	CBCSS39	Data Analytics with HADOOP	MATLAB Computations	Web Based Programming
NOTE:- Relevant CBCS courses, of minimum 3 credits, may also be chosen from other departments.				

Detail - Syllabi

Section-1: Core Courses

CSC11: Digital Logic and Computer Architecture	Prerequisite Course (if any): None

LEARNING OUTCOMES

Understanding of Boolean Algebra and Simplification of Boolean Functions
Understanding the digital logic gates, Combinational Logic and Sequential Logic
Understanding of Microprocessor Architecture and Micro-operations
Understanding of CPU and Binary Arithmetic

1. **Information Representation:** Number Systems - Binary, Octal, Decimal, and Hexa-Decimal; Number Base Conversions; Binary Arithmetic; Complements: (r-1)'s Complement, r's Complement, Subtraction using Complements; Floating Number-Fixed-point Representation, Floating-point Representation; Binary Codes for Decimal Digits: BCD Code, Excess-3 Code, 84-2-1 Code, 2421 Code, Reflected Code; Error Detection Code; Character Representation – ASCII, EBCDIC.
2. **Boolean Algebra, Logic Gates and simplification**: Boolean Algebra-Basic Definitions, Postulate, Basic Theorems and Properties of Boolean Algebra; Boolean Functions, Canonical and Standard Forms: Minterms and Maxterms, SOP, POS Conversion Between Canonical Forms, Standard Form of a Boolean Function; Other Logical Operations; Digital Logic Gates, Implementation of Boolean Functions, Simplification using boolean Algebra and Karnaugh Maps (K-Map) Method.
3. **Cobinational and Sequential Logic Circuit:** Overview of Combinational Logic; Combinational Logic Design Procedure; Design of Some Standard Combinational Circuits: Half Adder, Full Adder, Half Subtractor, Full Subtractor, Code Conversion; Decimal Adder, BCD Adder, Magnitude Comparator, Decoders, Encoder, Multiplexers, De-multiplexer, Flip-Flops: RS Flip Flop, Clocked RS, JK Flip Flop, Master Slave JK Flip Flop, D Type Flip Flop, T Type Flip Flop, State Table, State Diagram, State Equations, Flip Flop Characteristic Tables; Flip Flop Excitation Tables; Design of Sequential Circuits.
4. **Register Transfer and Micro Operations:** Register Transfer Language (RTL); Register Transfer; Bus Transfer; Memory Transfers; Arithmetic Microoperations; Logic Microoperations, List of Logic Microoperations, Addressing Modes, Data Transfer.
5. **Central Processing Unit (CPU):** Introduction; General Register Organization; Control Word; Stack Organization – Register Stack, Memory Stack, Reverse Polish Notation, Evaluation of Arithmetic Expression. Instruction Format – Three Address Instructions, Two Address Instructions, One Address Instructions, Zero Address Instructions. Parallel Processing; Pipelining – Arithmetic Pipeline, Instruction Pipeline,

REFERENCES

Mansaf Alam & Bashir Alam: Digital Logic Design. PHI
M. Morris Mano: Digital Logic and Computer Design. Pearson
M. Morris Mano: Computer System Architecture. Pearson
William Stalling: Computer Organization and Architecture. Prentice Hall
V. Rajaraman & T. Radhakrishnan: Computer Organization and Architecture. PHI
Donald D. Givone: Digital Principles and Design. McGraw Hill

CSC12: Discrete Mathematics	Prerequisite Course (if any): None

LEARNING OUTCOMES

Comprehend the values of basic structures in computer science and their algorithmic utility & significance.
Illustrate and apply logic propositional and predicate logic in decision problems.
Appreciate the context and utility of relations and posets in computer science.
Solve practical problems of complexity analysis by counting.
Illustrate the typical graph techniques and their applications.

1. **Basic Structures**: Set, Multi-set and Sequences; Type of sets, Set Operations, Power Set, Cartesian Products, Relation, Representation of relation, composition of relations, Functions, Types of Functions, Inverse of a functions, Compositions of functions, function representation, Sequences, Special Integer Sequences, Summations.
2. **Relations and Partial Orders:** Equivalence Relation, Reflexive, Symmetric and Transitive Closure, Transitive Closure and Warshall's Algorithm; Equivalence Classes and Partitions; Partial Ordering, Lexicographic Order, Hasse Diagram, Maximal and Minimal Elements, Lattices; Algebraic structure: Semi group, monoid.
3. **Logic, Reasoning and Inferences:** Foundations of Logic; Propositions, Conditional Statements, Bi- conditionals, Truth Table, Precedence of Logical Operators, Translating English Sentences, System Specifications, Logic Puzzles, Propositional, Logical Equivalences, De Morgan's Laws, and Construction of New Logical Equivalences; Predicates and Quantifiers – Predicates, Quantifiers, Universal Quantifiers, Quantifiers with Restricted Domains, Precedence, Binding Variables, Logical Equivalences, Negating Quantified Expressions, Translation to English Expressions, Rules of Inferences, valid Arguments, Proof Techniques.
4. **Counting:** Simple and Complex Counting Problems, Inclusion-Exclusion Principle; Tree Diagrams; The Pigeonhole Principle; Permutations, Combinations, Binomial Coefficients, Examples and Applications; Binomial Coefficients, Binomial Theorem, Expression, and other Identities; Permutations and Combinations with Repetition, Permutations with Indistinguishable Objects, Distributing Events into Boxes; Generating Permutations and Combinations.
5. **Recurrences:** homogeneous linear recurrences, Non-homogeneous linear recurrences, Solving recurrences using induction method, Solving recurrences using characteristic equations, Solving recurrences by domain transformation, Solving recurrences by substitution.

REFERENCES

K. Rosen: Discrete Mathematics and its Applications with Combinatorics and Graph Theory. McGraw Hill
Bernard Kolman, Robert Busby, and Sharon C. Ross: Discrete Mathematical Structures. Prentice Hall
J.P. Tremblay, and R. Manohar: Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill

CSC13: Algorithmics and Program Design	Prerequisite Course (if any): None

LEARNING OUTCOMES

Illustrate algorithmic terminology and issues; and analyze the efficiency of algorithms.
Develop iterative/recursive flowcharts and algorithms for basic computational problems.
Devise factoring algorithms, analyze and develop their improved versions.
Implement and analyze different array-based searching and sorting algorithms.
Design well-styled programs, trace and test the same.

1. **Algorithmic Problem Solving**: Algorithms; Problem Solving Aspect: Algorithm Devising, Design and Top-down Design; Algorithm Implementation: Essential and Desirable Features of an Algorithm; Efficiency of an Algorithm, Analysis of Algorithms, Pseudocodes; Algorithm Efficiency, Analysis and Order; Importance of Developing Efficient Algorithms; Complexity Analysis of Algorithms: Every-Case Time Complexity, Worst-Case Time Complexity, Average-Case Time Complexity, Best-Case Time Complexity, Flowchart. Flowchart – Symbols and Conventions.
2. **Basic Algorithms** – Exchanging the Values of Two Variables, Counting, Summation of a Set of Numbers, Factorial Computation, Sine Function Computation, Generation of the Fibonacci Sequence, Reversing the Digits of an Integer, Base Conversion, etc., Recursive Algorithms.
3. **Factoring:** Finding the square root of number, Smallest Divisor of an integer, Greatest common divisor of two integers, generating prime numbers, computing prime factors of an integer, Generation of pseudo random numbers, Raising a number to a large power, Computing the *n*th Fibonacci number.
4. **Arrays, Searching and Sorting:** Single and Multidimensional Arrays, Array Order Reversal**,** Array counting, Finding the maximum/minimum number in a list, Efficient algorithm for finding max-min in a list, partitioning an array, Monotones Subsequence; Searching: Linear and Binary Array Search, interpolation search; Sorting: Sorting by selection, Exchange and Insertion. Sorting by diminishing increment, Sorting by partitioning.
5. **Programming:** Introduction, Game of Life, Programming Style: Names, Documentation and Format, Refinement and Modularity; Coding, Testing and Further Refinement: Stubs and Drivers; Program Tracing, Testing, Evaluation; Program Maintenance: Program Evaluation, Review, Revision and Redevelopment; and Problem Analysis, Requirements Specification, Coding and Programming Principles.

REFERENCES

R.G. Dromy: How to Solve by Computer. Pearson (Unit 1-4)
R. Kruse, C.L. Tondo, B. Leung, and S. Mogalla: Data Structures and Program Design in C. Pearson (Unit-5)
L.A. Robertson: Simple Program Design, A Step-by-Step Approach. Thomson

CSC14: Database Management System	Prerequisite Course (if any): ….

LEARNING OUTCOMES

Fundamentals of databases with their advantages and functionalities
Various database models. Details of RDBMS, its advantages and usage
ER Diagrams and EER Diagrams
Normalization and it various usage in table optimization

1. **Basic Concepts:** Data, Database and DBMS; Database vs. Traditional File System Approach; Three Schema Architecture of DBMS, Data Independence; Categories of Database Management Systems: Hierarchical, Network and Relational Database Systems.
2. **Database Models**: Introduction, Categories of Database Models: High-level or Conceptual Data Models, Representational or Implementation Data Models, Low-level or Physical Data Models, Object Data Models. Entity relationship (ER) Model: Basic Concepts and their representations – Entity, Entity Type and Entity Set, Attributes and Keys, Relationships, Relationship Types, and Structural Constraints, Weak Entity, Naming Conventions & Design Issues in ER Model. ER and EER Diagrams.
3. **Relational Database Model:** Structure of Relational Model; Domains, Attributes, Tuples, and Relations; Characteristics of Relations; Relational Constraints – Domain Constraints, Key Constraints, Entity Integrity, and Referential Integrity Constraints; Relational Database Schema; Relational Algebra Operations – Select, Project, Rename, Union, Intersection, Set Difference, Join, and Division Operations; Aggregate Functions and Groupings.
4. **Structured Query Language (SQL):** Schema, Table and Domain Creation; Schema and Table Deletion; Table Modification; Insert, Delete, and Update Statements; SELECT- FROM-WHERE Structure; Renaming Attributes; Nested Queries and Set Comparisons; EXISTS and UNIQUE Functions; Aggregate Functions; Creating and Updating Views. Introduction to PL/SQL.
5. **Functional Dependencies and Normalization:** Informal Design Guidelines for Relation Schemas; Functional Dependencies; Inference Rules for Functional Dependencies; Normalization using Functional Dependencies – First Normal Form (INF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF); Multi- Valued Dependencies and Fourth Normal Form (4NF); Join Dependencies and Fifth Normal Form (5NF); Relation Decomposition and Insufficiency of Normal Forms; Dependency Preserving and Lossless Join Decompositions; Null Values and Dangling Tuples. Transaction Management and Concurrency Control: Transaction Concept; Transaction State; Concurrent Executions; Serializability and Recoverability; Testing for Serializability. Concurrency Control – Lock-Based Protocols and Timestamp-Based Protocols.

REFERENCES

R. Elmasri, S. B. Navathe: Fundamentals of Database Systems. Pearson

A. Silberschatz, H. F. Korth, and S. Sudarshan: Database System Concepts. McGraw Hill
J. Casteel: ORACLE 9i Developer: PL/SQL Programming. Thomson
Ivan Bayross: SQL, PL/SQL The Programming Language of of Oracle. BPB

CSC15: Lab-I (APD)

Implementation of at least ONE specific assignment concerning each of the following:

1. Derive the worst case, best case and average case time complexity for searching and sorting algorithms.
2. Generate the Fibonacci series using iterative and recursive functions.
3. Implement Euclids division algorithms for calculating GCD.
4. Implement an algorithm to generate pseudo random numbers.
5. Implement merge and Quick sort algorithms and determine the time required to sort the elements.
6. Implement binary search to find an element in a sorted array.
7. Design and implement an algorithm to search for the maximum and minimum element in an array.
8. Implement interpolation searching.

CSC16: Lab-II (DBMS)

Implementation of at least ONE specific assignment concerning each of the following:

1. SQL statements to create, update, and delete databases, tables, and views.
2. SQL statements to insert, update, and delete records from tables.
3. Simple SQL queries to retrieve information from a database.
4. Nested SQL queries to handle complex information retrieval requirements.
5. Managing changes affecting the data using COMMIT, ROLLBACK and SAVEPOINT.
6. Providing security to databases using GRANT and REVOKE commands.
7. SQL queries using order by, group by and having clauses and SQL sub queries, nested queries, join operations.
8. PL/SQL blocks using basic data types, branching and looping constructs.
9. Database triggers, functions/procedures and packages using PL/SQL.

Section-3.1: CBCS Courses (Sem-I)

CBCSE17.1: Statistical Computing	**Prerequisite Course (if any): ….**

LEARNING OUTCOMES

To apply discrete and continuous probability distributions to various business problems.
To perform Test of Hypothesis as well as calculate confidence interval and understand the concept of p-values.
To learn non-parametric test such as the Chi-Square test for Independence as well as Goodness of Fit.
To compute Bivariate and Multivariate Regression and Correlation and perform ANOVA and F-test.

1. **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.
2. **Solution of Equations and System of Simultaneous Equations:** Solution of Algebraic and Transcendental Equations using Bisection, Regula False, and Newton Raphson Methods, Gauss Elimination, Gauss Seidel, and Jacobi Methods.
3. **Interpolation, Numerical Differentiation and Integration, and Differential Equations:** Interpolation using Lagrange, and Newton's methods, Extrapolation, Least Square Fitting, Numerical Integration using Trapezoidal, and Simpson's Rules, Numerical Solution of Ordinary Differential Equations using Euler's and Range-Kutta Methods.
4. **Statistics:** Population, Sample, Sample Collection Methods, Data Representations and Classification, Central Tendency and Dispersion: Mean, Geometric Mean, Harmonic Mean, Median and Mode, Quartiles and Percentiles, Measures of Dispersion: Range, Variance, Standard Deviation, and Coefficient of Variation.
5. **Probability and Hypothesis Testing:** Sample Space, Events, Equally Likely Events, Probability, Independent Events, Addition and Multiplication Rules, Conditional Probability, Probability Distributions – Normal, Binomial, and Poison Distributions; Correlation using Karl Pearson and Spearman Method; Hypothesis Testing: t-Test, Chi-Square Test, Analysis of Variance (ANOVA), F-Test.

REFERENCES

S.C. Chapra & R.P.Canale: Numerical Methods for Engineering. TMH
V. Rajaraman: Computer oriented numerical methods. PHI

Andy Field, J. Miles, and Z. Field: Discovering Statistics Using R. SAGE
A. S. Grewal: Higher Engineering Mathematics. Khanna
M. K. Jain, S. R. K. Iyengav, and R. K. Jain: Numerical Methods for Scientific and Engineering Computation. New Age

CBCSE17.2: e-Business Systems	Prerequisite Course (if any): None

LEARNING OUTCOMES

Explain Systems; and IT related terminology, trends, challenges, obstacles and prospects.
Illustrate and perform work-centered analysis on business systems.
Devise factoring algorithms, analyze and develop their improved versions.
Implement and analyze different array-based searching and sorting algorithms.
Design well-styled programs, trace and test the same.

1. **Systems and IT:** Systems, Business System, Phases in Building and Maintaining Systems, IT-Based Innovations in Every Business Function, Product Design Systems, CAD Software, Procurement Systems, Supply Chain Management, Electronic Data Interchange, Manufacturing , Sales and Marketing Systems, Delivery Systems, Customer Service Systems, Finance Systems, Dramatic Progress in Processing Data, Recent Trends in IT; Applying IT to the Real World.
2. **Business Systems:** Frameworks and Models, Viewing Businesses as Systems, Businesses as Systems Consisting of Business Processes, The Value Chain , Business Processes and Functional Areas of Business , Information Systems and Work Systems, Increasing Overlap Between Information Systems and Work Systems, Framework for Thinking About Any System in Business , Work-Centered Analysis Framework, Five Perspectives for Viewing a Work System , Architecture: System Components and How They Operate Together, Performance: How Well the System Operates, Analyzing an IT-Enabled System From a Business Professional's Viewpoint, Work-Centered Analysis Method, Limitations and Pitfalls
3. **Business Processes and Models**: Business Processes, Process Modeling: DFD, Flowcharts and Structured English, Process Characteristics: Degree of Structure, Range of Involvement, Level of Integration, Rhythm, Complexity, Degree of Reliance on Machines etc, Communication and Decision Making; Evaluating Business Process Performance: Activity Rate and Output, Consistency, Productivity, Cycle Time, Downtime and Security, Basic Communication and Decision Making Concepts.
4. **Typical Information Systems:** Information System Categories related to Specific Functional Areas of Business, IS Categories applicable Functional Areas; Office Automation Systems; Communication Systems: Teleconferencing, E-Mail, Fax, SMS, Groupware, Internet, Intranets, Extranets, Knowledge Management, and Group Support Systems, Transaction Processing Systems, MIS and Executive Information Systems, Decision Support Systems, Case Based Reasoning, and Intelligent Systems, Enterprise Systems, Limitation and Uses of Typical Information Systems.
5. **Customer, Product and IT:** Customer's View of Product and services, The Customers' Experience, Evaluating Products and Services: Cost, quality, responsiveness, Reliability and Conformance to standards, Performance Variables of IT: Functional Capabilities and Limitations, Ease of use, Compatibility and Maintainability, Approaches of Organizational Computing-Centralized, Personal, Distributed, Networked and Client-Server, Current Limits of Software; Types of Software, Programming viewed as Business Process, Major Developments in Programming – Special purpose, Spreadsheets and CASE, Artificial Intelligence and Intelligent Systems.

REFERENCES

Steven Alter: Information Systems – The Foundations of E-Business. Pearson
S. Haag & M. Cummings: Information Systems Essentials. Mcgraw-Hill
R. C. Nickeson: Business Information Systems. Prentice Hall

CBCSE17.3: Human Computer Interaction	Prerequisite Course (if any): Computer Fundamentals

LEARNING OUTCOMES

Acquire first-hand experience of useful HCI techniques in practice.
Comprehend the latest topics in multimedia, global information systems, and the web-based models for rich interaction.
Articulate social and contextual models and theories related to HCI design processes,
Appreciate and illustrate the role of interaction design, universal access, and rich interaction.
Apply design principles and established rules during user-friendly interface designs.

1. **HCI Overview:** Need, Issues in Human Computer Interaction and Significance; Overview of Human Sensory Capabilities and Limitations: Input-Output Channels and Design Focus; Analysis of Design Experiments; Human Sensory Capabilities and Limitations: Memory, Types of Memory, Comparative association with Computer memory, Thinking and Emotions; Human Sensory Capabilities and Limitations: Individual Differences and Psychology of Design of Interactive Systems.
2. **Computers**: Text Entry, Design Focus- Numeric Pads, Positioning, Pointing and Display Devices; Display Devices, Devices for Virtual Reality and 3D interactions, Physical Controls, Sensors and Special Devices; Paper: Printing and Scanning, Readability of Text, Font Issues etc.
3. **Interaction:** Models of Interaction, Framework for HCI and Ergonomics; Industrial Interfaces; Interaction Styles: Introduction and Types with Examples, Navigation, Elements of WIMP Interface, Interactivity and its context, Experience and edutainment etc; Interaction Paradigms: Introduction, Different Paradigms of Interaction, Timesharing, Batch Processing, Personal computing, Distributed etc; Language vs Action; Hypertext,

Multimodality, WWW, Sensor based Context-aware Interactions etc.

4. **Interaction Design Basics:** Introduction, Design Basics and Process of Design, Scenarios, Navigation Design: Local structure, Global Structure, Dialog etc; Interaction Design Basics: Introduction, User Focus, Cultural Probes and Scenarios; Screen Design, Alignment and Layout, Screen Colors etc.
5. **Interaction Design Rules**: Introduction, Principles to Support Usability: Learn ability, Flexibility, and Robustness; Design Rules: Standards, need, significance; Underlying Theory, Usability factors etc.; Design Guidelines: Fundamental Guidelines, Golden Rules and Heuristics; Sheiderman Eight Rules; Norman's seven Principles for Transformation; HCI Patterns

REFERENCES

Alan Dix et al.: Human-Computer Interaction. Pearson

Carlo et al.: Human Computer Interaction. PHI

B. J. Mosley & B. A. Posey: Just Enough Software Test Automation. Prentice Hall