*B.Tech.*Computer Science and Engineering(CSE)

**KURUKSHETRA UNIVERSITY, KURUKSHETRA**

**MODIFIED SCHEME OF EXAMS W.E.F THE SESSION 2024-25**

## SEMESTER-III

| S. No. | Course No./ Code | Subject | L:T:P | Hours/ Week | Credits | Examination Schedule (Marks) | | | | Duration of Exam(Hours) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | End Semester Exam | Internal assessment | Practical Exam | Total | |
| 1 | B23-CSE-201 | Data Structures and Algorithms | 4:0:0 | 4 | 4 | 70 | 30 | 0 | 100 | 3 |
| 2 | B23-CSE-203 | Computer Organization & Architecture | 3:0:0 | 3 | 3 | 70 | 30 | 0 | 100 | 3 |
| 3 | B23-CSE-205 | Object Oriented Programming | 4:0:0 | 4 | 4 | 70 | 30 | 0 | 100 | 3 |
| 4 | B23-CSE-207 | IT Workshop (Python) | 3:0:0 | 3 | 3 | 70 | 30 | 0 | 100 | 3 |
| 5 | B23-BSC-203 | Probability and Statistics | 3:0:0 | 3 | 3 | 70 | 30 | 0 | 100 | 3 |
| 6 | B23-HSM-201 | Organizational Behaviour | 3:0:0 | 3 | 3 | 70 | 30 | 0 | 100 | 3 |
| 7 | B23-CSE-209 | Data Structures and Algorithms Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 8 | B23-CSE-211 | Object Oriented Programming Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 9 | B23-CSE-213 | IT Workshop (Python) Lab | 0:0:2 | 2 | 1 | 0 | 40 | 60 | 100 | 3 |
| 10 | B23-MAC-202 | Essence of Indian Traditional Knowledge | 2:0:0 | 2 | 1 | -- | 100 | -- | 100 | 3 |
| **TOTAL** | | | | **30** | **25** | **420** | **400** | **180** | **1000** | |

**Note:**

- NCC/NSS/Sports/Yoga/Technical/cultural club/society activities may also be joined by students in second year and will be evaluated in 7th semester by the institute based upon continuous evaluation model as per guidelines.

*B.Tech.* Computer Science and Engineering(CSE)

**KURUKSHETRA UNIVERSITY, KURUKSHETRA**

**MODIFIED SCHEME OF EXAMS W.E.F THE SESSION 2024-25**

## SEMESTER-IV

| S. No. | Course No./ Code | Subject | L:T:P | Hours/ Week | Credits | Examination Schedule (Marks) | | | | Duration of Exam(Hours) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | End Semester Exam | Internal assessment | Practical Exam | Total | |
| 1 | B23-CSE-202 | Advanced Programming (Java) | 4:0:0 | 4 | 4 | 70 | 30 | 0 | 100 | 3 |
| 2 | B23-ESC-212 | Digital Electronics | 4:0:0 | 4 | 4 | 70 | 30 | 0 | 100 | 3 |
| 3 | B23-CSE-204 | Design and Analysis Algorithms | 4:0:0 | 4 | 4 | 70 | 30 | 0 | 100 | 3 |
| 4 | B23-CSE-206 | Principles of Programming Languages | 4:0:0 | 4 | 4 | 70 | 30 | 0 | 100 | 3 |
| 5 | B23-HSM-302 | Intellectual Property Rights (IPR) and Regulatory | 3:0:0 | 3 | 3 | 70 | 30 | 0 | 100 | 3 |
| 6 | B23-CSE-208 | Advanced Programming Lab (Java) | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 7 | B23-ESC-214 | Digital Electronics Lab | 0:0:2 | 2 | 1 | 0 | 40 | 60 | 100 | 3 |
| 8 | B23-CSE-212 | Design & Analysis Algorithms Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 9 | B23-MAC-201 | Environmental Studies | 3:0:0 | 3 | 1 | 70 | 30 | 0 | 100 | 3 |
| **TOTAL** | | | | **30** | **24** | **420** | **300** | **180** | **900** | |

**Note:**

All students have to undertake the industrial training for 6 to 8 weeks after 4th semester which will be evaluated in 5th semester.

| B23-CSE-201 | Data Structures and Algorithms | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal assessment** | **Total** | **Time** |
| **4** | **0** | **0** | **4.0** | **70** | **30** | **100** | **3 Hour** |
| **Purpose** | To introduce the principles and paradigms of Data Structures for design and implement the software systems logically and physically. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO 1** | To introduce the basic concepts of Data structure, basic data types, searching and sorting based on array data types. | | | | | | |
| **CO 2** | To introduce the structured data types like Stacks and Queue and its basic operations's implementation. | | | | | | |
| **CO 3** | To introduce dynamic implementation of linked list. | | | | | | |
| **CO 4** | To introduce the concepts of Tree and graph and implementation of traversal algorithms. | | | | | | |

## UNIT-1

**Introduction to Data Structures**, Data Types, Built in and User Defined Data Structures, Applications of Data Structure, Algorithm Analysis, Worst, Best and Average Case Analysis, Notations of Space and Time Complexity, Basics of Recursion.

**Arrays**, One Dimensional Arrays, Two Dimensional Arrays and Multi-Dimensional Arrays, Sparse Matrices, Searching from array using Linear and Binary Searching Algorithm, Sorting of array using Selection, Insertion, Bubble sort.

## UNIT-2

**Stacks**: Definition, Implementation of Stacks and Its Operations, Evaluation of Infix, prefix and Postfix Expression, Inter-conversion of Infix, Prefix and Post-Fix Expression, Implementation of Merge Sort and Quick Sort Algorithm.

**Queues**: Definition, Sequential Implementation of Linear Queues and Its Operations, Circular Queue and its Implementation, Priority Queues and Its Implementation, Applications of queues.

## UNIT-3

**Linked Lists**: Need of Dynamic Data Structures, Single Link List and Its Dynamic Implementation, Traversing, Insertion, Deletion Operations on Single Link Lists. Comparison between Static and Dynamic, Implementation of Linked List.

Circular Link Lists and Doubly Link List, Dynamic Implementation of Primitive Operations on Doubly Linked Lists and Circular Link List, Dynamic Implementation of Stacks and Queues.

## UNIT-4

**Trees**: Definition, Basic Terminology, Binary Tree, External and Internal Nodes, Static and Dynamic Implementation of a Binary Tree, Primitive Operations on Binary Trees, Binary Tree Traversals: Pre-Order, In-Order and Post-Order Traversals, Representation of Infix, Post-Fix and Prefix Expressions using Trees.

Introduction to Binary Search Trees**:** B+ trees, AVL Trees, Threaded Binary trees, Balanced Multi-way search trees, Implementation of Heap Sort Algorithm.

**Graphs**: Basic Terminology, Definition of Undirected and Directed Graphs, Memory Representation of Graphs, Graph Traversals Algorithms: Breadth First and Depth First.

**Suggested Books:**


- Theory and Problems of Data Structures by Jr. Symour Lipschetz, Schaum's outline, TMH.
- Data Structures and Algorithms by PAI, TMH.
- Fundamentals of Data structures by Ellis Horowitz and Sartaj Sahni, Pub, 1983, AW.
- Data Structures and Algorithms by A.V. Aho, J.E. Hopcroft and T.D. Ullman, Original edition, Addison- Wesley, 1999, Low Priced Edition.
- Data Structures and Program Design in C by Robert Kruse, PHI,
- Shukla, Data Structures using C++, Wiley India
- Introduction to Computers Science -An Algorithms Approach , Jean Paul Tremblay, Richard B. Bunt, 2002, T.M.H.
- Data Structure and the Standard Template library – Willam J. Collins, 2003, T.M.H.

| B23-CSE-203 | Computer Organization & Architecture | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal assessment** | **Total** | **Time** |
| 3 | 0 | 0 | 3 | 70 | 30 | 100 | 3Hrs. |
| **Purpose** | Student will be able to understand the basic concepts of computer architecture and organization, and understand the key skills of constructing cost-effective computer systems. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Be familiar with the Computer arithematic and data representation | | | | | | |
| **CO2** | Be familiar with the basic computer organization and design | | | | | | |
| **CO3** | Be familiar with instruction set architecture and parallel processing. | | | | | | |
| **CO4** | Be acquainted with the basic knowledge of I/O organization. | | | | | | |

**UNIT- 1**

**Data representation and Computer arithmetic**: Introduction to Computer Systems, Organization and Architecture, Von Neumann Architecture, evolution and computer generations; fixed point, Floating-point and Decimal arithmetic operations, Digital arithmetic algorithms for Addition, Subtraction, Multiplication using Booth's algorithm, Multiprocessors and Multicomputer, MIPS, MFLOPS.
Memory Organization: Memory Hierarchy, Types of Memory, TLB

**UNIT-2**

**Basic Computer organization and Design**: General register organization, stack organization and common bus system, computer instructions, timing and control, Input, output and Interrupt: Interrupt cycle, Design drivers: common case, Amdahl's law.

Micro programmed Control organization, Control Memory, address sequencing, micro instruction format, Horizontal Vs Vertical micro-programming, design of control Unit, micro program sequencer, Hardwired v/s Micro-programmed. CISC and RISC: features and comparison.

**UNIT-3**

**Instruction set Architecture:** Instruction codes, instruction formats (Zero, One, Two and Three Address Instruction).Instruction cycle, reference instructions; Memory reference instructions.Various addressing modes.
Pipeline and vector Processing, Parallel Processing, Flynn's Taxonomy, Pipelining,
Instruction Pipeline, Basics of vector processing and Array Processors.

**UNIT-4**

**Input-output organization:** I/O interface. I/O Bus and interface modules, I/O versus Memory Bus.
Asynchronous data transfer: Strobe control, Handshaking, Asynchronous serial transfer. Modes of Transfer: Programmed I/O, Interrupt driven I/O, Priority interrupt; Daisy chaining, Parallel Priority interrupt. Direct memory Access, DMA controller and transfer. Input output Processor, CPU-IOP communication, Serial communication.

**Suggested Books:**


- William Stallings, “Computer Organization and Architecture – Designing for Performance”, Sixth Edition, Pearson Education, 2003.
- Morris Mano, M., “Computer System Architecture,” 3/e, Pearson Education, 2005.
- John P. Hayes, “Computer Architecture and Organization,” 3/e, TMH, 1998.
- David A. Patterson and John L. Hennessy, “Computer Organization and Design: The Hardware/Software interface”, Third Edition, Elsevier, 2005.
- V.P. Heuring, H.F. Jordan, “Computer Systems Design and Architecture”, Second Edition, Pearson Education, 2004.
- Carl Hamacher, ZvonkoVranesic and SafwatZaky, “Computer Organization”, Fifth Edition, Tata McGraw Hill, 2002.

| B23-CSE-205 | Object Oriented Programming | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Minor Test** | **Total** | **Time** |
| 4 | 0 | 0 | 4.0 | 70 | 30 | 100 | 3 Hour |
| **Purpose** | To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To introduce the basic concepts of object oriented programming language and the its representation. | | | | | | |
| **CO2** | To allocate dynamic memory, access private members of class and the behavior of inheritance and its implementation. | | | | | | |
| **CO3** | To introduce polymorphism, interface design and overloading of operator. | | | | | | |
| **CO4** | To handle backup system using file, general purpose template and handling of raised exception during programming. | | | | | | |

**UNIT–1**

Introduction to C++, C++ Standard Library, Illustrative Simple C++ Programs. Header Files, Namespaces, Application of object oriented programming. Control flow, variables and assignments statements, conditional execution, looping, function calls including recursion.

Object Oriented Concepts, Introduction to Objects and Object Oriented Programming, Encapsulation, Polymorphism, Overloading, Inheritance, Abstract Classes, Accessifier (public/ protected/ private), Class Scope and Accessing Class Members, Constant, Class Member, Structure and Class.Macro vs Inline Functions.

**UNIT-2**

Friend Function and Friend Classes, This Pointer, Dynamic Memory Allocation and Deallocation (New and Delete), Static Class Members, Constructors, parameter Constructors and Copy Constructors, Deconstructors,
Introduction of inheritance, Types of Inheritance, Overriding Base Class Members in a Derived Class, Public, Protected and Private Inheritance, Effect of Constructors and Deconstructors of Base Class in Derived Classes.

**UNIT-3**

Polymorphism, Pointer to Derived class, Virtual Functions, Pure Virtual Function, Abstract Base Classes, Static and Dynamic Binding, Virtual Deconstructors.

Fundamentals of Operator Overloading, Rules for Operators Overloading, Implementation of Operator Overloading Like <<,>> Unary Operators, Binary Operators.

**UNIT-4**

Text Streams and binary stream, Sequential and Random Access File creation and updation, Stream Input/ Output Classes, Stream Manipulators.

Basics of C++ Exception Handling, Try, Throw, Catch, multiple catch, Re-throwing an

Exception, Exception specifications.
Templates: Function Templates, Overloading Template Functions, Class Template, Class Templates and Non- Type Template arguments.

**Suggested Books:**


- The complete reference C ++ by Herbert shieldt Tata McGraw Hill.
- Object Oriented Programming in Turbo C++ by Robert Lafore, 1994, The WAITE Group Press.
- Shukla, Object Oriented Programming in c++, Wiley India.
- C++ How to Program by H M Deitel and P J Deitel, 1998, Prentice Hall.
- Programming with C++ By D Ravichandran, 2003, T.M.H.

| B23-CSE-207 | IT Workshop (Python) | | | | | | |
|---|---|---|---|---|---|---|---|
| **L** | **T** | **P** | **Credit** | **End Semester Exam** | **Internal assessment** | **Total** | **Time** |
| 3 | 0 | 0 | 3.0 | 70 | 30 | 100 | 3 Hours |
| **Purpose** | To familiarize the students with the basics of Python Programming | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO1** | To Study Fundamental concept of Python. | | | | | | |
| **CO2** | To Study and implement expression and Strings methods | | | | | | |
| **CO3** | To Study and implement tuples , list and dictionary operations. | | | | | | |
| **CO4** | To Study and implement exception handling and file operation. | | | | | | |

**Unit-1**

**Familiarization with the basics of Python programming**: Introduction to Python, Features of Python, Execution modes: interactive mode and script mode, Python character set, use of indentation, Python tokens( keyword, identifier, literal, operator, punctuator), variables, use of comments, Knowledge of data types: Number(Integer, Floating point, Complex).
**Errors**: syntax errors, logical errors, and run-time errors

**Unit-2**

**Expressions**: Statement, Type conversion, and input/output: Precedence of Operators,Arithmetic operators, relational operators, logical operators, assignment operators, augmented assignment operators, identity operators (is, is not),Expression, evaluation of an expression, type-conversion, Flow of Control,Conditional statements, Iterative Statements
**Strings**: Introduction, string operations (concatenation, repetition, membership and slicing), traversing a string using loops, built-in functions/methods–len(), capitalize(), title(), lower(), upper(), count(), find(), index(), endswith(), startswith(), isalnum(), isalpha(), isdigit(), islower(), isupper(), isspace(),lstrip(), rstrip(), strip(), replace(), join(), partition(), split()

**Unit-3**

**Array**: Access the Elements of an Array, Length of an Array, Adding Array Elements, Removing Array Elements, Adds and remove the element at the specified position. **Lists, Tuples, Dictionary**: introduction, indexing, list operations, traversing a list using loops, built-in functions/methods–len(), list(), append(), extend(), insert(), count(), index(), remove(), pop(), reverse(), sort(), sorted(), min(), max(), sum().
**Introduction to Python modules**: Importing module using 'import ' and using from statement, importing math module (pi, e, sqrt(), ceil(), floor(), pow(), fabs(), sin(), cos(), tan()); random module (random(), randint(), randrange()), statistics module (mean(), median(), mode()).Functions and its types (Built-in Functions, Functions defined in Module, User Defined Functions), arguments, default parameters, positional parameters, Function Returning Value(s), Recursion, Scope of a Variable.

**Unit-4**

**Files**: Introduction to files, types of files (Text file, Binary file, CSV file),Text file: opening a text file, file open modes (r, r+, w, w+, a, a+ etc), closing a text file, opening a file using with clause, writing/appending data to a text file using write() and writelines(), reading from a text file using read(), readline() and readlines

**Reference Book:**

1.The Complete Reference Python By Martin C Brown Publication by McGraw Hill.
2. Let us Python By Yashwant Kanetkar

| B23-BSC-203 | PROBABILITY AND STATISTICS | | | | | | |
|---|---|---|---|---|---|---|---|
| **L** | **T** | **P** | **Credit** | **End Semester Exam** | **Internal assessment** | **Total** | **Time** |
| **3** | **0** | **-** | **3.0** | **70** | **30** | **100** | **3 Hrs.** |
| **Purpose** | To familiarize the prospective students with the fundamentals of probability & statistics and how to apply the principles to model and analyze various phenomena in fields like finance, economics, and engineering, aiding in making informed decisions and predicting outcomes. | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO1** | introduce the fundamental concepts of probability to analyze and predict outcomes in real-life situations. | | | | | | |
| **CO 2** | obability theory provides models of probability distributions (theoretical models of the observable reality involving chance effects) to be tested by statistical methods which has various engineering applications. | | | | | | |
| **CO 3** | make the students familiar about basic statistics to analyze data sets using various measures of central tendency and dispersion | | | | | | |
| **CO 4** | on completion of Unit IV, students will proficiently apply correlation and regression techniques, including calculating coefficients and determining lines of regression, to analyze relationships between variables in datasets. | | | | | | |

**UNIT-I (10 Hrs)**

**Basic Probability:** Introduction, additive law of probability, Conditional Probability, Independent Events, Bayes' Theorem.

Random Variables: Discrete random variables, probability distribution, Probability mass function and distribution function, Expectation, Moments, Variance and standard deviation of discrete random variables.

**UNIT-II(12hrs)**

**Continuous Probability distribution:**

Continuous random variables, probability distribution, Probability density function and distribution function, Expectation, Moments, Variance and standard deviation of Continuous random variables.

Probability distributions: Binomial, Poisson and Normal - evaluation of statistical parameters for these three distributions.

**UNIT-III(10hrs)**

**Basic Statistics:**

Measures of Central tendency: Mean, median, quartiles, mode, Geometric mean, Harmonic mean, Measures of dispersion: Range, Quartile deviation, mean deviation, standard deviation, coefficient of variation, Moments, Skewness and Kurtosis.

**UNIT-IV(08hrs)**

**Correlation & Regression:**

Introduction,Correlation, Coefficient of correlation, methods of calculations, Lines of regression, Rank correlation.

**Suggested Books:**

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley &

Sons,2006.
2. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall, 2003 (Reprint).
3. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India, 2002.
4. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Ed.,Wiley, 1968.
5. N.P. Bali and and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
6. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
7. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.
8.Veerarajan T., Engineering Mathematics (for semester III), Tata McGraw-Hill, New Delhi, 2010.

| B23-HSM-201 | Organizational Behavior | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal assessment** | **Total** | **Time** |
| 3 | 0 | - | 3.0 | 70 | 30 | 100 | 3 Hours |
| **Course Outcomes** | | | | | | | |
| **Purpose** | The objective of this Course is to make students conversant with the basic concepts of organization behaviour for nurturing managerial skills. | | | | | | |
| **CO1** | An overview about organizational behavior as a discipline and understanding the concept of individual behavior. | | | | | | |
| **CO2** | Understand the concept and importance of personality, emotions and its importance in decision making and effective leadership. | | | | | | |
| **CO3** | Enabling the students to know about the importance of effective motivation and its contribution in group dynamics and resolving conflicts. | | | | | | |
| **CO4** | Understand how to overcome organizational stress by maintaining proper organizational culture and effective communication. | | | | | | |

## UNIT- 1

**Introduction to organizational behavior**: Concept and importance of organizational behavior, role of Managers in OB, challenges and opportunities for OB.

**Foundation of individual behavior**: Biographical characteristics, concept and types of abilities , concept of values and attitude, types of attitude, attitude and workforce diversity.

## UNIT- 2

**Introduction to personality and emotions**: Definition and Meaning of Personality, Determinants of Personality, Personality Traits Influencing OB, Nature and Meaning of Emotions, Emotions dimensions, concept of Emotional intelligence.

**Perception and individual decision making:** meaning of perception, factors influencing perception, rational decision making process, concept of bounded rationality. Leadership-trait approaches, behavioural approaches, situational approaches, and emerging approaches to leadership.

## UNIT-3

**Motivation:** Concept and theories of motivation, theories of motivation-Maslow, two factor theory, theory X and Y, ERG Theory, McClelland's theory of needs, goal setting theory, application of theories in organizational scenario, linkage between MBO and goal setting theory.

**Foundations of group behaviour and conflict management**: Defining and classifying of groups, stages of group development, Informal and formal groups- group dynamics, managing conflict and negotiation , causes of group conflicts, managing intergroup conflict through resolution.

## UNIT-4

**Introduction to Organizational Communication:** Meaning and importance of communication process, importance of effective communication, organizational stress: definition and meaning sources and types of stress, impact of stress on organizations, stress management techniques.

**Introduction to Organization Culture:** Meaning and nature of organization culture, types of culture, managing cultural diversity, managing change and innovation-change at work, resistance to change, a model for managing organizational change.

**Text Books**:

1. Colquitt, Jason A., Jeffery A. LePine, and Michael Wesson. Organizational Behavior: Improving Performance and Commitment in the Workplace. 5th ed. New York: McGrawHill Education, 2017.

2. Hitt, Michael A., C. Chet Miller, and Adrienne Colella. Organizational Behavior. 4th ed. Hoboken, NJ: John Wiley, 2015.

3. Robbins, Stephen P., and Timothy Judge. Organizational Behavior. 17th ed. Harlow, UK: Pearson Education, 2017. Stephen P. Robins, Organisational Behavior, PHI Learning / Pearson Education, 11th edition, 2008.

**Reference Books:**

1. Schermerhorn, Hunt and Osborn, Organisational behavior, John Wiley.

2. Udai Pareek, Understanding OrganisationalBehaviour, Oxford Higher Education.

3. Mc Shane & Von Glinov, OrganisationalBehaviour, Tata Mc Graw Hill.

4. Aswathappa, K., OrganisationalBehaviour– Text and Problem, Himalaya Publication.

| B23-CSE-209 | Data Structures and Algorithms Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Practical Exam** | **Total** | **Time** |
| 0 | 0 | 3 | 1.5 | 40 | 60 | 100 | 3 |
| **Purpose** | To introduce the principles and paradigms of Data Structures for design and implement the software systems logically and physically. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To introduce the basic concepts of Data structure, basic data types, searching and sorting based on array data types. | | | | | | |
| **CO2** | To introduce the structured data types like Stacks and Queue and its basic operation's implementation. | | | | | | |
| **CO3** | To introduce dynamic implementation of linked list. | | | | | | |
| **CO4** | To introduce the concepts of Tree and graph and implementation of traversal algorithms. | | | | | | |

1. Write a program for Binary search methods.
2. Write a program for insertion sort, selection sort and bubble sort.
3. Write a program to implement Stack and its operation.
4. Write a program for quick sort.
5. Write a program for merge sort.
6. Write a program to implement Queue and its operation.
7. Write a program to implement Circular Queue and its operation.
8. Write a program to implement singly linked list for the following operations: Create, Display, searching, traversing and deletion.
9. Write a program to implement doubly linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
10. Write a program to implement circular linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
11. Write a program to implement insertion, deletion and traversing in B tree

| B23-CSE-211 | Object Oriented Programming Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Practical Exam** | **Total** | **Time** |
| 0 | 0 | 3 | 1.5 | 40 | 60 | 100 | 3 Hour |
| **Purpose** | To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To introduce the basic concepts of object oriented programming language and the its representation. | | | | | | |
| **CO2** | To allocate dynamic memory, access private members of class and inheritance,Constructors. | | | | | | |
| **CO3** | To introduce polymorphism, interface design and overloading of operator. | | | | | | |
| **CO4** | To handle backup system using file, general purpose template and handling of raised exception during programming. | | | | | | |

**Q1**. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power
( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

**Q2.** Write a C++ program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class. b) Write a Program to Invoking Derived Class Member Through Base Class Pointer.

**Q3**. Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be 'Y' or 'N'. Some sample interaction with the program might look like this.

Enter first number, operator, and second number: 10/ 3 Answer = 3.333333
Do another (Y/ N)? Y
Enter first number, operator, second number 12 + 100 Answer = 112
Do another (Y/ N) ? N

**Q4**. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:

- Enter your area code, exchange, and number: 415 555 1212
- My number is (212) 767-8900
- Your number is (415) 555-1212

**Q5**. Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results maybe a DM object or DB objects, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and cenitmetres depending on the object on display.

**Q6**. Create a class rational which represents a numerical value by two double values- NUMERATOR and DENOMINATOR. Include the following public member Functions:

- constructor with no arguments (default).
- constructor with two arguments.
- void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
- Overload + operator to add two rational number.
- Overload >> operator to enable input through cin.
- Overload << operator to enable output through cout. Write a main ( ) to test all the functions in the class.

**Q7**. Consider the following class definition class father {

protected : int age; public;

father (int x) {age = x;} virtual void iam( )

{ cout<< "I AM THE FATHER, my age is : "<< age<< end1:}

};

Derive the two classes son and daughter from the above class and for each, define iam( ) to write our similar but appropriate messages. You should also define suitable constructors for these classes. Now, write a main ( ) that creates objects of the three classes and then calls iam ( ) for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam( ) through the pointer to demonstrate polymorphism in action.

**Q8**. Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name ( a string of 30 or lesser no. of characters) and marks.

**Q9**. A hospital wants to create a database regarding its indoor patients. The information to store include

a) Name of the patient
b) Date of admission
c) Disease
d) Date of discharge

Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

**Q10**. Create a class **Employee** with a name and salary. Create a class **Manager** inherit from **Employee**. Add an instance variable, named department, of type string. Supply a method to **to String** that prints the manager's name, department and salary. Create a class **Executive** inherits from **Manager**. Supply a method **to String** that prints the string **"Executive"** followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and methods.

**Q11**. Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the ESC kay should cause the program to print out the total cars and total cash and then exit.

**Q12**. Write a function called reversit( ) that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to reversit( ) as an argument. Write a program to exercise reversit( ). The program should get a string from the user, call reversit( ), and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba)".

**Q13**. Create a class Student with a name and roll no. as data member. Create a Class Template for student class. The program should also implement template overloading.

**Q14**. Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class account that stores customer name, account number and type of account. From this derive the classes

cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

a) Accept deposit from a customer and update the balance.
b) Display the balance.
c) Compute and deposit interest.
d) Permit withdrawal and update the balance.
e) Check for the minimum balance, impose penalty, necessary and update the balance.
f) Do not use any constructors. Use member functions to initialize the class members.

| B23-CSE-213 | IT Workshop (Python) Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **L** | **T** | **P** | **Credit** | **Internal assessment** | **Practical Exam** | **Total** | **Time** |
| 0 | 0 | 2 | 1 | 40 | 60 | 100 | 3 Hours |
| **Purpose** | The course is designed to provide Basic knowledge of Python. | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO1** | To study fundamentals of python programming and implement basic programs. | | | | | | |
| **CO2** | To implement the searching technique using python. | | | | | | |
| **CO3** | To implement sorting techniques using python. | | | | | | |
| **CO4** | To implement matrix multiplication using python. | | | | | | |

**LIST OF PROGRAMS**

1. Write a program to compute the GCD of two numbers.
2. Write a program to find the square root of a number
3. Write a program to find the Exponentiation (power of a number)
4. Write a program to find the maximum of a list of numbers
5. Write a program for Linear search and Binary search
6. Write a program for Selection sort
7. Write a program for Insertion sort
8. Write a program to find first n prime numbers
9. Write a program to multiply matrices
10. Write a program that take command line arguments (word count)
11. Write a program to find the most frequent words in a text read from a file

| B23-MAC-202 | Essence of Indian Traditional Knowledge | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Internal Assessment** | **Practical Exam** | **Total** | **Time** |
| **2** | **-** | **-** | **1.0** | **100** | **-** | **100** | **3** |
| **Purpose** | To facilitate the students with the concepts of Indian traditional knowledge and to make them understand the importance of roots of knowledge system, analyse and apply to their day to day life. | | | | | | |
| | **Course Outcomes** | | | | | | |
| **CO 1** | The students will be able to understand, connect up and explain basics of Indian traditional knowledge in modern scientific perspective | | | | | | |
| **CO2** | The students will be able to understand Holistic Health using Indian Knowledge System | | | | | | |
| **CO3** | The students will be able to manage thoughts and emotions, will learn positivity, self regulation and control | | | | | | |
| **CO4** | The students will be able to achieve Consciousness through Indian Knowledge System | | | | | | |

**UNIT-1**

Introduction to Indian Traditional knowledge: Define traditional knowledge, importance, kinds of traditional knowledge. Philosophical systems, Basics of Rajyoga and Karamyoga, Benefits of Rajyogaand Karamyoga.

**UNIT-2**

Holistic Health using Indian Knowledge System:
Basic principles of natural life style, Benefits through five elements. Healing through food, Chakras and Mudras.
Physical, Mental, Emotional and Spiritual health using traditional knowledge .

**UNIT-3**

Positivity: Traditional approaches. Happiness: objective and subjective measures of wellbeing, life satisfaction. Resilience, Self-regulation and self-control, optimism, self-esteem. Managing thoughts and Emotions with the help of Rajyoga. Achieving Powers for Self Mastery.

**UNIT-4**

Achieving Consciousness through Indian Knowledge System: Emotional intelligence, Indian approach to Psychology. Consciousness; levels, body-mind relationship, self motivation, Self and Identity in modern Psychology and Indian thought., Spirituality and well being.

**Reference and Text Books:**

Mahadevan, M., Bhat, V.R. & Pavana N. (2022). Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning

- Baumgardner, SR & Crothers, MK (2009). Positive Psychology. Prentice Hall/Pearson Education.
- Cornelissen, R.M., Misra G. & Varma S. (2014). Foundations & Applications of Indian Psychology. Pearson Education.

Rajyoga Education and Consciousness Improvement Programme for Educators, Rajyoga Education and Research Foundation.

Rajyoga Meditation Course, Thoughkart, Jaipur(Rajasthan), India.

PrakartikSwasthya Shastra, Publisher Natural Lifestyle

| B23-CSE-202 | Advanced Programming (Java) | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal Assessment** | **Total** | **Time** |
| 4 | - | - | 4.0 | 70 | 30 | 100 | 3 hrs |
| **Purpose** | To introduce the concepts of advanced java and its implementations. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Study fundamental concepts of Java. | | | | | | |
| **CO2** | To study and implement String and Collection methods. | | | | | | |
| **CO3** | To study oops concept and implement abstraction, inheritance | | | | | | |
| **CO4** | To study servlet and data base connectivity of java and java servlets. | | | | | | |

## UNIT – 1

Introduction: Importance and features of Java, Concepts of Java Virtual machine (JVM), Java Comments, Keywords, Constants, Variables and Data types, java Type Casting, Wrapper classes, Operators and Expressions, Control Statements, Conditional Statements, Loops and Iterations. Creating an Array of one and two dimensional . Java Math methods. Method parameter, Calling Methods.

## UNIT – 2

String, String Buffer and string Builder classes. String methods (char At, concat, compare To, equals, get Chars, length, replace, to Lower Case, to Upper Case etc. ).

Java Collection: list interface(Array, vector, stack, linked list),Queue interface, Map interface, Tree Set and Tree Map interface, Hash Map and Hash Set interface.

Exception Handling, Manual Exception creation, File Handling

## UNIT – 3

Class definition, adding Variables and Methods, creating Objects, Constructors, java Modifier, java encapsulation, java inheritance, method Overloading and Over Riding, Java abstraction, Java interface, Java Packages.

## UNIT – 4

JDBC: JDBC Fundamentals, Establishing Connectivity and working with connection interface, working with statements, Creating and Executing SQL statements(creation of table, insertion, deletion, updation).

Servlets: Introduction to Servlets, Life cycle of Servlets, Creating, Compiling and running Servlet, Reading the servlet Parameters, Reading Initialization parameter, Handling HTTP Request and Response (GET / POST Request), Session Tracking.

**Suggested Books:**

1. Gary Cornell and Horstmann Cay S., Core Java, Vol I and Vol II, Sun Microsystems Press.

2. Herbert Schildt, Java: The Complete Reference, McGraw-Hill.
3. Philip Hanna, JSP: The Complete Reference, McGraw-Hill.
4. Deital and Deital, Java How to Program, Prentice Hall (2007).

| B23-ESC-212 | Digital Electronics | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal Assessment** | **Total** | **Time** |
| 4 | - | - | 4.0 | 70 | 30 | 100 | 3 Hour |
| **Purpose** | To learn the basic methods for the design of digital circuits and provide the fundamental concepts used in the design of digital systems. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To introduce basic postulates of Boolean algebra and shows the correlation between Boolean expressions | | | | | | |
| **CO2** | To introduce the methods for simplifying Boolean expressions | | | | | | |
| **CO3** | To outline the formal procedures for the analysis and design of combinational circuits and sequential circuits | | | | | | |
| **CO4** | To introduce the concept of memories and programmable logic devices. | | | | | | |

## UNIT-1
## MINIMIZATION TECHNIQUES AND LOGIC GATES

Binary Digits, Logic Levels, and Digital Waveforms, Logic Systems-Positive and negative, Logic Operations, Logical Operators, Logic Gates-AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR, Active high and Active low concepts, Universal Gates and realization of other gates using universal gates, Gate Performance Characteristics and Parameters. Boolean Algebra: Rules and laws of Boolean algebra, Demorgan's Theorems, Boolean Expressions and Truth Tables, Standard SOP and POS forms; Minterm and Maxterms, Canaonical representation of Boolean expressions, Duality Theorem, Simplification of Boolean Expressions, Minimization Techniques for Boolean Expressions using
Karnaugh Map. Introduction of TTL and CMOS Logic and their characteristics,
Tristate gates.

## UNIT-2
## COMBINATIONAL CIRCUITS

Introduction to combinational Circuits, Adders-Half-Adder and Full-Adder, Subtractors- Half and Full Subtractor;
Parallel adder and Subtractor; Look-Ahead Carry Adders. BCD adder, BCD subtractor, Parity Checker/Generator, Multiplexer, Demultiplexer, Encoder, Priority Encoder; Decoder ,BCD to Seven segment Display Decoder/Driver, and Comparators.

## UNIT-3
## SEQUENTIAL CIRCUITS

Introduction to Sequential Circuits, Flip-Flops: Types of Flip Flops -RS, T, D, JK; Edge triggering, Level Triggering; Flip Flop conversions; Master-Salve JK.

Introduction to shift registers, Basic Shift Register Operations, types of shift registers, Bidirectional Shift Registers,
Shift Register Counters. Introduction to counters, Types of Counters-Asynchronous and

synchronous counters, Up/Down Synchronous Counters, Modulo-n Counter , State table, excitation table concepts, Design of asynchronous and synchronous counters, Ring Counter, Applications of counters.

## UNIT-4
## CONVERTER and MEMORY DEVICES

Digital to Analog Converter, Weighed Register: R-2R Ladder Network: Analog to Digital Conversion, Successive Approximation Type, Dual Slope Type.

Classification of memories - ROM: ROM organization, PROM, EPROM, EEPROM, EAPROM, RAM: - RAM organization - Write operation, Read operation, Memory cycle, Timing wave forms, memory expansion, Static RAM Cell, MOSFET RAM cell structure, Dynamic RAM cell structure, Programmable Logic Devices - Programmable Logic Array (PLA), Programmable Array Logic (PAL), Implementation of PLA, PAL using ROM.

**Suggested Books:**


- Donald P. Leach and Albert Paul Malvino, Digital Principles and Applications, 8th Edition, TMH, 2003.M.
- Morris Mano, Digital Design, 3rd Edition, Prentice Hall of India Pvt. Ltd., 2003 / Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2003.
- ALI, Digital Switching Systems, , TMH
- A.K. Maini, Digital Electronics, Wiley India
- John F. Wakerly, Digital Design, Fourth Edition, Pearson/PHI, 2006
- John. M Yarbrough, Digital Logic Applications and Design, Thomson Learning, 2002.
- S. Salivahanan and S. Arivazhagan, Digital Circuits and Design, 3rd Edition., Vikas Publishing House Pvt. Ltd, New Delhi, 2006
- William H. Gothmann, Digital Electronics, 2nd Edition, PHI, 1982.
- Thomas L. Floyd, Digital Fundamentals, 8th Edition, Pearson Education Inc, New Delhi, 2003 □ Donald D. Givone, Digital Principles and Design, TMH, 2003.

| B23-CSE-204 | Design and Analysis of Algorithms | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal Assessment** | **Total** | **Time** |
| 4 | - | - | 4.0 | 70 | 30 | 100 | 3 Hrs. |
| **Purpose** | To introduce advanced data structures & algorithms concepts involving their implementation for solving complex applications. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Learn the basic concepts of data structures and their analysis. | | | | | | |
| **CO2** | Study the concept of dynamic programming and various advanced data structures. | | | | | | |
| **CO3** | Learn various graph algorithms and concepts of computational complexities. | | | | | | |
| **CO4** | Study various String matching algorithms | | | | | | |

## UNIT-1
## Introduction

**Review** :- Elementary Data Structures, Algorithms & its complexity(Time & Space), Analysing Algorithms, Asymptotic Notations, Pseudocode Conventions, Binary search trees.

**Recurrence relation**:- Methods for solving recurrence(Substitution , Recursion tree, Master theorem).

## UNIT-2
## Advanced Design and analysis Techniques

**Dynamic programming**:- Elements, Matrix-chain multiplication, longest common subsequence,

**Greedy algorithms**:- Elements , Activity- Selection problem, Huffman codes, Task scheduling problem, Knapsack problem.

**Backtracking algorithms**:- Graph coloring, N-Queen problem, Hamiltonian path and circuit.

## UNIT-3
## Graph Algorithms

**Review of graph algorithms:-**Traversal Methods(Depth first & Breadth first search),Topological sort, Strongly connected components, Minimum spanning trees- Kruskal's and Prim's Algorithm, Single source shortest paths, Relaxation, Dijkstra's Algorithm, Bellman-Ford algorithm, Single source shortest paths for directed acyclic graphs, Floyd-Warshall algorithm,

## UNIT-4
## Computational Complexity

Basic Concepts, Polynomial vs Non-Polynomial Complexity, NP- hard & NP-complete classes. The Naïve string-matching algorithm, Rabin-Karp Algorithm, String matching with finite automata.

**Text Books:**


1. Corman, Leiserson and Rivest : Introduction to Algorithms, 2/e, PHI
2. Harsh Bhaisn, Algorithms: Design And Analysis Oxford University Press,2015.

**Reference Books:**


1. Aho, Hopcroft and Ullman : The Design and Analyses of Computer Algorithms. Addison Wesley.
2. R.B.Patel, Expert Data Structures with C, Khanna Publications , Delhi, India, 2ndEdition 2004, ISBN 81-87325-07-0, pp.1-909.
3. R.B.Patel& M.M.S Rauthan, Expert Data Structures with C++, Khana Publications, Delhi , India, 2ndEdition 2004,ISBN : 87522-03-8.
4. Horowitz, Ellis and Sahni, Sartaj : Fundamentals of Computer Algorithms, Galgotia Publications

| B23-CSE-206 | Principles of Programming Languages | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal Assessment** | **Total** | **Time** |
| 4 | - | - | 4.0 | 70 | 30 | 100 | 3 Hour |
| **Purpose** | To introduce the principles and paradigms of programming languages for design and implement the software intensive systems. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO 1** | To introduce the basic concepts of programming language, the general problems and methods related to syntax and semantics. | | | | | | |
| **CO 2** | To introduce the structured data objects, subprograms and programmer defined data types. | | | | | | |
| **CO 3** | To outline the sequence control and data control. | | | | | | |
| **CO 4** | To introduce the concepts of storage management using programming languages. | | | | | | |

## UNIT-1
### Introduction, Syntax and Semantics

**Introduction:** A brief history, Characteristics of a good programming language, Programming language translators- compiler and interpreters, Elementary data types – data objects, variable and constants, data types. Specification and implementation of elementary data types, Declarations, type checking and type conversions, Assignment and initialization, Numeric data types, enumerations, Booleans and characters.

**Syntax and Semantics:** Introduction, general problem of describing syntax, Formal method of describing Syntax, attribute grammar dynamic semantic.

## UNIT-2
### Structured data objects, Subprograms and Programmer Defined Data Types

**Structured data objects:** Structured data objects and data types, specification and implementation of structured data types, Declaration and type checking of data structure, vector and arrays, records Character strings, variable size data structures, Union, pointer and programmer defined data objects, sets, files.

**Subprograms and Programmer Defined Data Types:** Evolution of data type concept abstraction, encapsulation and information hiding, Subprograms, type definitions, abstract data types, over loaded subprograms, generic subprograms.

## UNIT–3
### Sequence Control and Data Control

**Sequence Control:** Implicit and explicit sequence control, sequence control within expressions, sequence control within statement, Subprogram sequence control: simple call return, recursive subprograms, Exception and exception handlers, co routines, sequence control. Concurrency – subprogram level concurrency, synchronization through semaphores, monitors and message passing

**Data Control:** Names and referencing environment, static and dynamic scope, block structure, Local data and local referencing environment, Shared data: dynamic and static scope, Parameter and parameter transmission schemes.

## UNIT-4

## Storage Management and Programming Languages

**Storage Management:** Major run time elements requiring storage, programmer and system controlled storage management and phases, Static storage management, Stack based storage management, Heap storage management, variable and fixed size elements.

**Programming Languages:** Introduction to procedural, non-procedural, structured, logical, functional and object oriented programming language, Comparison of C and C++ programming languages.

**Suggested Books:**


- Terrence W. Pratt, Marvin V. Zelkowitz, Programming Languages Design and Implementation, Pearson.
- Allen Tucker and Robert Noonan, Programming Languages–Principles and Paradigms, Tata McGraw-Hill,2009.
- Ellis Horowitz, Fundamentals of Programming Languages, Galgotia Publications, 2010.
- C. Ghezzi, Programming Languages Concepts, Wiley Publications, 2010.

| B23-HSM-302 | Intellectual Property Rights (IPR) & Regulatory | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **End Semester Exam** | **Internal Assessment** | **Total** | **Time** |
| 3 | - | - | 3.0 | 70 | 30 | 100 | 3 Hours |
| **Course Outcomes** | | | | | | | |
| **Purpose** | The course is designed to provide comprehensive knowledge to the students regarding the general principles of IPR, Concept and Theories, International Regime Relating to IPR | | | | | | |
| **CO1** | Students will be familiarized with the introduction about patent concept and legal implications | | | | | | |
| **CO2** | Students will be able understand the concept of copyright in detail | | | | | | |
| **CO3** | Students will be able to understand trademark and law associated with it | | | | | | |
| **CO4** | Students will be able to know about geographical Indications, industrial design and IPR in information Technology | | | | | | |

## UNIT-1

Indian patent law: The patents act, 1970,amendments to the patents act, patentable subject matter, patentability criteria, procedure for filing patent applications, patent granting procedure, revocation, patent infringement and remedies, relevant provisions of the biological diversity act, 2002, access and benefit sharing issues, objectives, rights, patent act 1970 and its amendments. procedure of obtaining patents, working of patents. infringement.

## UNIT-2

Copyrights: Introduction, works protected under copyright law, infringement. introduction to copyright, international protection of copyright and related rights- an overview Indian copyright act, 1957 with its amendments, copyright works, ownership, transfer and duration of copyright, renewal and termination of copyright

Industrial Designs : Need for protection of industrial designs, subject matter of protection and requirements, the designs act, 2000,procedure for obtaining design protection, revocation, infringement and remedies.

## UNIT-3

Trademarks : Objectives, types, rights, protection of goodwill, infringement, passing off, need for protection of trademark, kinds of trademark , Indian trademarks law, procedural requirements of protection of trademarks, content of the rights, exhaustion of rights, procedural requirements of protection of trademarks, content of the rights, exhaustion of rights, assignment under licensing, infringement, right of goodwill, passing off, domain names and effects of new technology (internet).

## UNIT-4

**Geographical Indications**: Objectives, Justification, International Position, Multilateral Treaties, National Level, Indian Position. Industrial Designs: Objectives, Rights, Assignments, Infringements, Information Technology Related Intellectual Property Rights, Computer Software and Intellectual Property, Database and Data Protection, Protection of Semiconductor chips, Domain Name Protection, Implications of intellectual property rights on the commercialization of Biotechnology products.

**References:**


- N.S. Gopalakrishnan & T.G. Agitha, Principles of Intellectual Property (2009), Eastern Book Company, Lucknow
- David I. Bainbridge, Intellectual Property, Longman, 9th Edition, 2012
- Susan K Sell, Private Power, Public Law: The Globalization of Intellectual Property Rights, Cambridge University Press, 2003
- N.S. Gopalakrishnan & T.G. Ajitha, Principles of Intellectual Property, Eastern Book Company,2nd Edition , 2014
- Jayashree Watal, Intellectual Property Rights in the WTO and Developing Countries, Oxford University Press, 2001
- Lionel Bently & Brad Sherman, Intellectual Property Law, Oxford University Press, 3rd Edition, 2008
- Duggal Pavan, Legal Framework on Electronic Commerce & Intellectual Property Rights, Universal Publishing House, 2014
- Paul Torremans, Intellectual Property And Human Rights, Kluwer Law International, 2008
- Steven D Anderman, Interface Between Intellectual Property Rights and Competition Policy, Cambridge University Press, 2007.
- Philippe Cullet, Intellectual Property Protection and Sustainable Development, Lexis Nexis, 2005

| B23-CSE-208 | Advanced Programming Lab (Java) | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Internal Assessment** | **Practical Exam** | **Total** | **Time** |
| -- | -- | 3 | 1.5 | 40 | 60 | 100 | 3 Hour |
| **Purpose** | To introduce the concepts of Advanced Java Programming | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Study fundamental concepts of Java. | | | | | | |
| **CO2** | To study and implement String and Collection methods. | | | | | | |
| **CO3** | To study oops concept and implement abstraction, inheritance | | | | | | |
| **CO4** | To study servlet and data base connectivity of java and java servlets. | | | | | | |

## List of Practicals

1. Write a Java program to implement matrix multiplication
2. Write a java program to implement String, String Buffer and String builder.
3. Write a java program to implement Stack and queue.
4. Write a java program to handle File Exception, Arithmetic exception and Array out of bound Exception.
5. Write a java program to throw user defined exception.
6. Write a java program to implement multiple inheritance
7. Write a java program to calculate area of cube and volume using abstraction.
8. Write a java program to insert and display data from database.
9. Write a java servlet program to display data from HTML form.
10. Write a java servlet program to setup session tracking using cookie.
11. Write a java servlet program to set session tracking using hidden form and url rewriting.
12. Write a java servlet program to display data from employee table.

| B23-ESC-214 | Digital Electronics Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Internal Assessment** | **Practical Exam** | **Total** | **Time** |
| 0 | 0 | 2 | 1.0 | 40 | 60 | 100 | 3 Hrs |
| **Purpose** | To learn the basic methods for the design of digital circuits and systems. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To Familiarization with Digital Trainer Kit and associated equipment. | | | | | | |
| **CO2** | To Study and design of TTL gates | | | | | | |
| **CO3** | To learn the formal procedures for the analysis and design of combinational circuits. | | | | | | |
| **CO4** | To learn the formal procedures for the analysis and design of sequential circuits | | | | | | |

**LIST OF EXPERIMENTS:**

1. Familiarization with Digital Trainer Kit and associated equipment.
2. Study of TTL gates AND, OR, NOT, NAND, NOR, EX-OR, EX-NOR.
3. Design and realize a given function using K-Maps and verify its performance.
4. To verify the operation of Multiplexer and De-multiplexer.
5. To verify the operation of Comparator.
6. To verify the truth table of S-R, J-K, T, D Flip-flops.
7. To verify the operation of Bi-directional shift register.
8. To design and verify the operation of 3-bit asynchronous counter.
9. To design and verify the operation of asynchronous Up/down counter using J-K FFs.
10. To design and verify the operation of asynchronous Decade counter.
11. Study of TTL logic family characteristics.
12. Study of Encoder and Decoder.
13. Study of BCD to 7 segment Decoder.

| B23-CSE-212 | Design and Analysis of algorithms Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Internal Assessment** | **Practical Exam** | **Total** | **Time** |
| -- | -- | 3 | 1.5 | 40 | 60 | 100 | 3 Hour |
| **Purpose** | The student will learn the algorithm analysis techniques, become familiar with the different algorithm design techniques and understand how to implement various algorithms | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | The student should be able to Design algorithms for various computing problems | | | | | | |
| **CO2** | The student should be able to Analyse the time and space complexity of algorithms. | | | | | | |
| **CO3** | The student should be able to critically analyse the different algorithm design techniques for a given problem. | | | | | | |
| **CO4** | The student should be able to modify existing algorithms to improve efficiency. | | | | | | |

## List of Practicals

1. Program to find the given element in Binary Search Tree.
2. Program to implement Binary Search Tree.
3. Program to implement Prim's algorithm using greedy method.
4. Program to implement Kruskal's algorithm using greedy method.
5. Program to implement graph traversal using Breadth First Search.
6. Program to implement graph traversal using Depth First Search.
7. Program to implement N queen's problem.
8. Program to implement all pairs shortest path.
9. Program to implement Activity Selection Problem.
10. Program to implement Knapsack problem.
11. Program to implement Graph Coloring Problem.
12. Program to implement Naïve String matching algorithm.

| B23-MAC-201 | Environmental Studies | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Internal Assessment** | **End Semester Exam** | **Total** | **Time** |
| 3 | -- | -- | 1 | 30 | 70 | 100 | 3 Hour |
| **Purpose** | | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Students will be able to understand the importance of natural resources. | | | | | | |
| **CO2** | Students will understand the concept of an ecosystem, its structure, and its functions. | | | | | | |
| **CO3** | The students will be able to understand the causes and impacts of various environmental pollution. | | | | | | |
| **CO4** | Students will be able to understand the relationship between human population and the environment. | | | | | | |

## UNIT-1

**Introduction to Environmental studies**: The Multidisciplinary nature of environmental studies Definition; Scope and importance, Need for public awareness.

**Natural Resources:** Forest resources: Use and Over-exploitation, deforestation. Timber extraction, mining, dams, and their effects, Water resources: Use and over-utilization of surface and groundwater, conflicts over water, dams benefits and problems, Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, Food resources: changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, Energy resources: renewable and non-renewable energy sources, Land resources: land degradation, soil erosion, and desertification.

## UNIT-2

**Ecosystems**: Concept of an ecosystem, Structure, and function of an ecosystem, Energy flow in the ecosystem, Ecological succession, Food chains, food webs, and ecological pyramids. Major types of ecosystem-Forest ecosystem, Grassland ecosystem, Desert ecosystem, Aquatic ecosystem.

**Biodiversity and its Conservation:** Introduction-Definition: genetic, species, and ecosystem diversity. Biogeographical classification of India, Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values, India as a mega-diversity nation, Hot-spots of biodiversity, Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, Endangered and endemic species of India, Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

## UNIT-3

**Environmental pollution**: Causes, effects, and control measures of: - Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Nuclear hazards, and Solid waste Management: Causes, effects, and control measures of urban and industrial wastes, Disaster management: floods, earthquake, cyclone and landslides.

**Social Issues and the Environment:** Sustainable development, Water conservation, rainwater harvesting, Resettlement and rehabilitation of people; its problems and concerns. Environmental

ethics: Issues and possible solutions, Climate change, global warming, acid rain, ozone layer depletion, and wasteland reclamation. Environment Protection Act, Air (Prevention and Control of Pollution) Act, Water (Prevention and Control of Pollution) Act, Wildlife Protection Act., and Forest Conservation Act.

**UNIT-4**

**Human population and the Environment:** Population growth, Population Explosion-Family welfare Programme, Environment and human health. Human Rights, Value Education, HIV/AIDS, Women and Child Welfare, Role of Information Technology in Environment and human health, Drugs and their effects; Useful and harmful drugs; Use and abuse of drugs; Stimulant and depressant drugs. Concept of drug de-addiction. Legal position on drugs and laws related to drugs.

**Field Work (Practical)-**

- Visit to a local area to document environmental assets -river/forest/grassland/ hill/mountain.
- Visit to a local polluted site- Urban/Rural/Industrial/Agricultural.
- Study of common plants, insects, and birds.
- Study of simple ecosystems- pond, river, hill slopes, etc.

**Suggested Readings:**


1. Agarwal, K.C. 2001 Environmental Biology, Nidi Publ. Ltd. Bikaner.
2. Kaushik, Anubha and Kaushik, C.P. (2004 Perspectives in Environmental Studies, New age International Publishers.
3. Bharucha Erach, The Biodiversity of India, Mapin Publishing Pvt. Ltd., Ahmedabad - 380013, India, Email: mapin@icenet. net (R).
4. Brunner R.C., 1989, Hazardous Waste Incineration, McGraw Hill Inc. 480p.
5. Clerk B.S., Marine Pollution, Clanderson Pross Oxford (TB).
6. Cunningham, W.P.Cooper, T.H. Gorhani, E & Hepworth, M.T.2001, Environmental Encyclopedia, Jaico Publ. House, Mumbai, 1196p.
7. De A.K., Environmental Chemistry, Wiley Eastern Ltd.
   Down to Earth, Centre for Science and Environment (R).