CSXXX	OBJECT ORIENTED PROGRAMMING
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Basic knowledge of any programming language. Preferably C.
Course Objective:	• Equip the students with the basic programming concepts. • Introduce different techniques pertaining to problem solving skills. • Arm the students with the necessary constructs of C++ programming.
Course Outcome:	On completion of the course, a student should be able to: • prepare object-oriented design for small/medium scale problems. • demonstrate the differences between traditional imperative design and object-oriented design • explain class structures as fundamental, modular building blocks • understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • write small/medium scale C++ programs with simple graphical user interface • use classes written by other programmers when constructing their systems • understand and to use fundamental data structures: lists, stacks, queues and trees
Syllabus	
Module I **[10] Hours**	**Module I** Introduction to object-oriented programming features: abstraction, encapsulation, polymorphism, inheritance. Getting started with C++ syntax, headers and name spaces, datatype, variables, stream I/O, constants, references, operators, flow control, arrays and pointers, C-types string, functions, default values in functions, structure, and union.

	Abstraction mechanism: Class overview, access control, class members, class scope, objects creation, inline function, function overloading, ambiguity in function overloading. Constructors, destructors, object as function argument, array of object, string, standard C++ string class. Constant members, Constant object, static class members, this pointer, friend functions.
Module II **[10] Hours**	**Module II** Dynamic memory allocation, new and delete operators, object copying, copy constructor, assignment operator, object pass by reference, and reference return. Operator overloading: Overloading unary operator, binary operator, member, and non-member operator function. Data conversion. Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, is-a vs has-a relationship, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors.
Module III **[10] Hours**	**Module III** Dynamic polymorphism: Late binding, Base class pointer, object slicing, method overriding with virtual functions, pure virtual functions, abstract classes, virtual destructor. Template: template classes, template functions. Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration. Namespaces: user defined namespaces, namespaces provided by library. File stream: Text file operation, binary file. STL: algorithms, containers, iterators.
Suggested Books:	1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India) 2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India) 3. The C++ Programming Language by Bjarne Stroustrup 4. Big C++ - Wiley India 5. Mastering C++ - Venugopal, McGraw-Hill Education (India) 6. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education 7. C++ and Object-Oriented Programming – Jana, PHI Learning.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	OBJECT ORIENTED PROGRAMMING LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic knowledge of any programming language. Preferably C.
Course Objective:	• Equip the students with the basic programming concepts. • Introduce different techniques pertaining to problem solving skills. • Arm the students with the necessary constructs of C++ programming.
Course Outcome:	On completion of the course, a student should be able to: • prepare object-oriented design for small/medium scale problems. • demonstrate the differences between traditional imperative design and object-oriented design • explain class structures as fundamental, modular building blocks • understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • write small/medium scale C++ programs with simple graphical user interface • use classes written by other programmers when constructing their systems • understand and to use fundamental data structures: lists, stacks, queues and trees
Syllabus	
Experiment 1	Programs on implementing abstraction mechanism using class and object.
Experiment 2	Programs on inline-function, default argument to function, function overloading, friend function, namespace.
Experiment 3	Programs on use of constructor, destructor, constant members, static class members

Experiment 4	Program on memory operation with new, delete operator, copy constructor, assignment operator
Experiment 5	Programs on operator overloading.
Experiment 6	Programs on inheritance, multilevel, and multiple inheritance, constructor, destructor execution in inheritance, virtual base class.
Experiment 7	Programs on dynamic polymorphism: virtual function
Experiment 8	Programs on generic programming, template function, template class
Experiment 9	Exception Handling
Experiment 10	File stream operation
Suggested Books:	1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India) 2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India) 3. The C++ Programming Language by Bjarne Stroustrup 4. Big C++ - Wiley India 5. Mastering C++ - Venugopal, McGraw-Hill Education (India) 6. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education 7. C++ and Object-Oriented Programming – Jana, PHI Learning.
Evaluation:	Number of experiments to be carried out = 10 Maximum marks per experiment = 6 Total = 6 × 10 = 60 Components of each experiment and their weights: • Attendance = 1 mark • Correct program execution = 4 marks • Timely record submission = 1 mark Written/ Viva / Project = 40 marks

CSXXX	DESIGN AND ANALYSIS OF ALGORITHMS (DAA)
Credits: 4 (3-1-0)	
Pre-requisite for this course:	• Basic of Data Structures and Programming knowledge
Course Objective:	The course objectives: • To understand the importance of algorithm and its complexity • To analyze the complexity of an algorithm in terms of time and space complexities • To design and implement various programming paradigms and its complexity
Course Outcome:	On completion of the course, a student should be able to: • Ability to analyze the time and space complexity, given an algorithm • Apply the techniques of algorithm in solving real world problems • Systematic development of an algorithm for solving a problem
Syllabus	
Module I **[10] Hours**	Introduction Role of Algorithm in computing Growth of Functions (Asymptotic notations, standard notations, and common functions) Best Case, Worst Case, Average Case Recurrences, solution of recurrences by substitution recursion tree and Master methods Design & Analysis of Divide and conquer algorithms, Quick sort, Heapsort: Heaps, Building a heap, The heapsort algorithm, Priority Queue, Binary search, Lower bounds for sorting.
Module II **[10] Hours**	Dynamic programming algorithms (Matrix-chain multiplication, Elements of dynamic programming, Longest common subsequence) Greedy Algorithms - (Activity- selection Problem, Elements of Greedy strategy, Fractional knapsac problem, Huffman codes).
Module III **[10] Hours**	Data structure for disjoint sets:- Disjoint set operations, Linked list representation, Disjoint set forests. Graph Algorithms: Graph and their Representations, Breadth first and depth-first search, Minimum Spanning Trees, Kruskal and Prim's algorithms, single-source shortest paths (Bellman-ford and Dijkstra's algorithms), All-pairs shortest paths (Floyd – Warshall Algorithm) ,Back tracking, Branch and Bound.

Module IV **[10] Hours**	Fast Fourier Transform, string matching (Rabin-Karp algorithm) NP–Completeness (Polynomial time, Polynomial time verification, NP - Completeness and reducibility, NP-Complete problems (without Proofs), Approximation algorithms (Vertex-Cover Problem, Traveling Salesman Problem).
Suggested Books:	1. Introduction to Algorithms, third edition, The MIT Press, (2009). Cormen, T H, C E Leiserson, R L Rivest, and C Stein. 2. Algorithms, Cengage Learning,(2008), Jerome L Paul, Kenneth A Berman. 3. Computer Algorithms: Introduction to Design & Analysis, third edition, Pearson Education, (2009), Sara Baase, Allen Van Gelder. 4. Fundamentals of Algorithm, 2nd Edition, Universities Press, (2008), Horowitz & Sahani. 5. Algorithm Design: Foundations, Analysis And Internet Examples, first edition, Wiley India,(2010), Goodrich, Tamassia.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	DESIGN AND ANALYSIS OF ALGORITHM LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objective is to: • Design the solution of a problem in an optimal way so that the time complexity and memory usage of the solution must be minimized.
Course Outcome:	On completion of the course, a student should be able to: • Systematic development of a solution to solve a problem. • Analyze the space requirement during implementation of the solution. • The overall time complexity of the solution.
Syllabus	For each lab provide the specific instruction(s) and questions required to carry out the lab experiments • Lab-1 Introduction • Lab-2 Recursion • Lab-3 Divide and Conquer approach • Lab-4 Divide and Conquer (Cont..) • Lab-5 Dynamic Programming • Lab-6 Greedy Approach • Lab-7 Disjoint Set data structure • Lab-8 Graph algorithms • Lab-9 Graph Algorithms (cont..) • Lab-10 Backtracking
Evaluation	1. Lab Assessment: 50% 2. End Term: 50%

CSXXX	OBJECT ORIENTED PROGRAMMING LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic knowledge of any programming language. Preferably C.
Course Objective:	• Equip the students with the basic programming concepts. • Introduce different techniques pertaining to problem solving skills. • Arm the students with the necessary constructs of C++ programming.
Course Outcome:	On completion of the course, a student should be able to: • prepare object-oriented design for small/medium scale problems. • demonstrate the differences between traditional imperative design and object-oriented design • explain class structures as fundamental, modular building blocks • understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • write small/medium scale C++ programs with simple graphical user interface • use classes written by other programmers when constructing their systems • understand and to use fundamental data structures: lists, stacks, queues and trees
Syllabus	
Experiment 1	Programs on implementing abstraction mechanism using class and object.
Experiment 2	Programs on inline-function, default argument to function, function overloading, friend function, namespace.
Experiment 3	Programs on use of constructor, destructor, constant members, static class members

Experiment 4	Program on memory operation with new, delete operator, copy constructor, assignment operator
Experiment 5	Programs on operator overloading.
Experiment 6	Programs on inheritance, multilevel, and multiple inheritance, constructor, destructor execution in inheritance, virtual base class.
Experiment 7	Programs on dynamic polymorphism: virtual function
Experiment 8	Programs on generic programming, template function, template class
Experiment 9	Exception Handling
Experiment 10	File stream operation
Suggested Books:	1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India) 2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India) 3. The C++ Programming Language by Bjarne Stroustrup 4. Big C++ - Wiley India 5. Mastering C++ - Venugopal, McGraw-Hill Education (India) 6. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education 7. C++ and Object-Oriented Programming – Jana, PHI Learning.
Evaluation:	Number of experiments to be carried out = 10 Maximum marks per experiment = 6 Total = 6 × 10 = 60 Components of each experiment and their weights: • Attendance = 1 mark • Correct program execution = 4 marks • Timely record submission = 1 mark Written/ Viva / Project = 40 marks

CSXXX	OBJECT ORIENTED PROGRAMMING
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Basic knowledge of any programming language. Preferably C.
Course Objective:	• Equip the students with the basic programming concepts. • Introduce different techniques pertaining to problem solving skills. • Arm the students with the necessary constructs of C++ programming.
Course Outcome:	On completion of the course, a student should be able to: • prepare object-oriented design for small/medium scale problems. • demonstrate the differences between traditional imperative design and object-oriented design • explain class structures as fundamental, modular building blocks • understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • write small/medium scale C++ programs with simple graphical user interface • use classes written by other programmers when constructing their systems • understand and to use fundamental data structures: lists, stacks, queues and trees
Syllabus	
Module I **[10] Hours**	**Module I** Introduction to object-oriented programming features: abstraction, encapsulation, polymorphism, inheritance. Getting started with C++ syntax, headers and name spaces, datatype, variables, stream I/O, constants, references, operators, flow control, arrays and pointers, C-types string, functions, default values in functions, structure, and union.

	Abstraction mechanism: Class overview, access control, class members, class scope, objects creation, inline function, function overloading, ambiguity in function overloading. Constructors, destructors, object as function argument, array of object, string, standard C++ string class. Constant members, Constant object, static class members, this pointer, friend functions.
Module II **[10] Hours**	**Module II** Dynamic memory allocation, new and delete operators, object copying, copy constructor, assignment operator, object pass by reference, and reference return. Operator overloading: Overloading unary operator, binary operator, member, and non-member operator function. Data conversion. Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, is-a vs has-a relationship, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors.
Module III **[10] Hours**	**Module III** Dynamic polymorphism: Late binding, Base class pointer, object slicing, method overriding with virtual functions, pure virtual functions, abstract classes, virtual destructor. Template: template classes, template functions. Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration. Namespaces: user defined namespaces, namespaces provided by library. File stream: Text file operation, binary file. STL: algorithms, containers, iterators.
Suggested Books:	1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India) 2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India) 3. The C++ Programming Language by Bjarne Stroustrup 4. Big C++ - Wiley India 5. Mastering C++ - Venugopal, McGraw-Hill Education (India) 6. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education 7. C++ and Object-Oriented Programming – Jana, PHI Learning.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	DESIGN AND ANALYSIS OF ALGORITHMS (DAA)
Credits: 4 (3-1-0)	
Pre-requisite for this course:	• Basic of Data Structures and Programming knowledge
Course Objective:	The course objectives: • To understand the importance of algorithm and its complexity • To analyze the complexity of an algorithm in terms of time and space complexities • To design and implement various programming paradigms and its complexity
Course Outcome:	On completion of the course, a student should be able to: • Ability to analyze the time and space complexity, given an algorithm • Apply the techniques of algorithm in solving real world problems • Systematic development of an algorithm for solving a problem
Syllabus	
Module I **[10] Hours**	Introduction Role of Algorithm in computing Growth of Functions (Asymptotic notations, standard notations, and common functions) Best Case, Worst Case, Average Case Recurrences, solution of recurrences by substitution recursion tree and Master methods Design & Analysis of Divide and conquer algorithms, Quick sort, Heapsort: Heaps, Building a heap, The heapsort algorithm, Priority Queue, Binary search, Lower bounds for sorting.
Module II **[10] Hours**	Dynamic programming algorithms (Matrix-chain multiplication, Elements of dynamic programming, Longest common subsequence) Greedy Algorithms - (Activity- selection Problem, Elements of Greedy strategy, Fractional knapsac problem, Huffman codes).
Module III **[10] Hours**	Data structure for disjoint sets:- Disjoint set operations, Linked list representation, Disjoint set forests. Graph Algorithms: Graph and their Representations, Breadth first and depth-first search, Minimum Spanning Trees, Kruskal and Prim's algorithms, single-source shortest paths (Bellman-ford and Dijkstra's algorithms), All-pairs shortest paths (Floyd – Warshall Algorithm) ,Back tracking, Branch and Bound.

Module IV **[10] Hours**	Fast Fourier Transform, string matching (Rabin-Karp algorithm) NP–Completeness (Polynomial time, Polynomial time verification, NP - Completeness and reducibility, NP-Complete problems (without Proofs), Approximation algorithms (Vertex-Cover Problem, Traveling Salesman Problem).
Suggested Books:	1. Introduction to Algorithms, third edition, The MIT Press, (2009). Cormen, T H, C E Leiserson, R L Rivest, and C Stein. 2. Algorithms, Cengage Learning,(2008), Jerome L Paul, Kenneth A Berman. 3. Computer Algorithms: Introduction to Design & Analysis, third edition, Pearson Education, (2009), Sara Baase, Allen Van Gelder. 4. Fundamentals of Algorithm, 2nd Edition, Universities Press, (2008), Horowitz & Sahani. 5. Algorithm Design: Foundations, Analysis And Internet Examples, first edition, Wiley India,(2010), Goodrich, Tamassia.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	OBJECT ORIENTED PROGRAMMING LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic knowledge of any programming language. Preferably C.
Course Objective:	• Equip the students with the basic programming concepts. • Introduce different techniques pertaining to problem solving skills. • Arm the students with the necessary constructs of C++ programming.
Course Outcome:	On completion of the course, a student should be able to: • prepare object-oriented design for small/medium scale problems. • demonstrate the differences between traditional imperative design and object-oriented design • explain class structures as fundamental, modular building blocks • understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • write small/medium scale C++ programs with simple graphical user interface • use classes written by other programmers when constructing their systems • understand and to use fundamental data structures: lists, stacks, queues and trees
Syllabus	
Experiment 1	Programs on implementing abstraction mechanism using class and object.
Experiment 2	Programs on inline-function, default argument to function, function overloading, friend function, namespace.
Experiment 3	Programs on use of constructor, destructor, constant members, static class members

Experiment 4	Program on memory operation with new, delete operator, copy constructor, assignment operator
Experiment 5	Programs on operator overloading.
Experiment 6	Programs on inheritance, multilevel, and multiple inheritance, constructor, destructor execution in inheritance, virtual base class.
Experiment 7	Programs on dynamic polymorphism: virtual function
Experiment 8	Programs on generic programming, template function, template class
Experiment 9	Exception Handling
Experiment 10	File stream operation
Suggested Books:	1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India) 2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India) 3. The C++ Programming Language by Bjarne Stroustrup 4. Big C++ - Wiley India 5. Mastering C++ - Venugopal, McGraw-Hill Education (India) 6. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education 7. C++ and Object-Oriented Programming – Jana, PHI Learning.
Evaluation:	Number of experiments to be carried out = 10 Maximum marks per experiment = 6 Total = 6 × 10 = 60 Components of each experiment and their weights: • Attendance = 1 mark • Correct program execution = 4 marks • Timely record submission = 1 mark Written/ Viva / Project = 40 marks

CSXXX	OBJECT ORIENTED PROGRAMMING
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Basic knowledge of any programming language. Preferably C.
Course Objective:	• Equip the students with the basic programming concepts. • Introduce different techniques pertaining to problem solving skills. • Arm the students with the necessary constructs of C++ programming.
Course Outcome:	On completion of the course, a student should be able to: • prepare object-oriented design for small/medium scale problems. • demonstrate the differences between traditional imperative design and object-oriented design • explain class structures as fundamental, modular building blocks • understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • write small/medium scale C++ programs with simple graphical user interface • use classes written by other programmers when constructing their systems • understand and to use fundamental data structures: lists, stacks, queues and trees
Syllabus	
Module I **[10] Hours**	**Module I** Introduction to object-oriented programming features: abstraction, encapsulation, polymorphism, inheritance. Getting started with C++ syntax, headers and name spaces, datatype, variables, stream I/O, constants, references, operators, flow control, arrays and pointers, C-types string, functions, default values in functions, structure, and union.

	Abstraction mechanism: Class overview, access control, class members, class scope, objects creation, inline function, function overloading, ambiguity in function overloading. Constructors, destructors, object as function argument, array of object, string, standard C++ string class. Constant members, Constant object, static class members, this pointer, friend functions.
Module II **[10] Hours**	**Module II** Dynamic memory allocation, new and delete operators, object copying, copy constructor, assignment operator, object pass by reference, and reference return. Operator overloading: Overloading unary operator, binary operator, member, and non-member operator function. Data conversion. Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, is-a vs has-a relationship, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors.
Module III **[10] Hours**	**Module III** Dynamic polymorphism: Late binding, Base class pointer, object slicing, method overriding with virtual functions, pure virtual functions, abstract classes, virtual destructor. Template: template classes, template functions. Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration. Namespaces: user defined namespaces, namespaces provided by library. File stream: Text file operation, binary file. STL: algorithms, containers, iterators.
Suggested Books:	1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India) 2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India) 3. The C++ Programming Language by Bjarne Stroustrup 4. Big C++ - Wiley India 5. Mastering C++ - Venugopal, McGraw-Hill Education (India) 6. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education 7. C++ and Object-Oriented Programming – Jana, PHI Learning.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	DESIGN AND ANALYSIS OF ALGORITHMS (DAA)
Credits: 4 (3-1-0)	
Pre-requisite for this course:	• Basic of Data Structures and Programming knowledge
Course Objective:	The course objectives: • To understand the importance of algorithm and its complexity • To analyze the complexity of an algorithm in terms of time and space complexities • To design and implement various programming paradigms and its complexity
Course Outcome:	On completion of the course, a student should be able to: • Ability to analyze the time and space complexity, given an algorithm • Apply the techniques of algorithm in solving real world problems • Systematic development of an algorithm for solving a problem
Syllabus	
Module I **[10] Hours**	Introduction Role of Algorithm in computing Growth of Functions (Asymptotic notations, standard notations, and common functions) Best Case, Worst Case, Average Case Recurrences, solution of recurrences by substitution recursion tree and Master methods Design & Analysis of Divide and conquer algorithms, Quick sort, Heapsort: Heaps, Building a heap, The heapsort algorithm, Priority Queue, Binary search, Lower bounds for sorting.
Module II **[10] Hours**	Dynamic programming algorithms (Matrix-chain multiplication, Elements of dynamic programming, Longest common subsequence) Greedy Algorithms - (Activity- selection Problem, Elements of Greedy strategy, Fractional knapsac problem, Huffman codes).
Module III **[10] Hours**	Data structure for disjoint sets:- Disjoint set operations, Linked list representation, Disjoint set forests. Graph Algorithms: Graph and their Representations, Breadth first and depth-first search, Minimum Spanning Trees, Kruskal and Prim's algorithms, single-source shortest paths (Bellman-ford and Dijkstra's algorithms), All-pairs shortest paths (Floyd – Warshall Algorithm) ,Back tracking, Branch and Bound.

Module IV **[10] Hours**	Fast Fourier Transform, string matching (Rabin-Karp algorithm) NP–Completeness (Polynomial time, Polynomial time verification, NP - Completeness and reducibility, NP-Complete problems (without Proofs), Approximation algorithms (Vertex-Cover Problem, Traveling Salesman Problem).
Suggested Books:	1. Introduction to Algorithms, third edition, The MIT Press, (2009). Cormen, T H, C E Leiserson, R L Rivest, and C Stein. 2. Algorithms, Cengage Learning,(2008), Jerome L Paul, Kenneth A Berman. 3. Computer Algorithms: Introduction to Design & Analysis, third edition, Pearson Education, (2009), Sara Baase, Allen Van Gelder. 4. Fundamentals of Algorithm, 2nd Edition, Universities Press, (2008), Horowitz & Sahani. 5. Algorithm Design: Foundations, Analysis And Internet Examples, first edition, Wiley India,(2010), Goodrich, Tamassia.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	DESIGN AND ANALYSIS OF ALGORITHM LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objective is to: • Design the solution of a problem in an optimal way so that the time complexity and memory usage of the solution must be minimized.
Course Outcome:	On completion of the course, a student should be able to: • Systematic development of a solution to solve a problem. • Analyze the space requirement during implementation of the solution. • The overall time complexity of the solution.
Syllabus	For each lab provide the specific instruction(s) and questions required to carry out the lab experiments • Lab-1 Introduction • Lab-2 Recursion • Lab-3 Divide and Conquer approach • Lab-4 Divide and Conquer (Cont..) • Lab-5 Dynamic Programming • Lab-6 Greedy Approach • Lab-7 Disjoint Set data structure • Lab-8 Graph algorithms • Lab-9 Graph Algorithms (cont..) • Lab-10 Backtracking
Evaluation	1. Lab Assessment: 50% 2. End Term: 50%

CSXXX	COMPUTER ORGANIZATION AND ARCHITECTURE
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Basic of Digital Electronics
Course Objective:	• To have a thorough understanding of the basic organization and architecture of a digital computer. • To study different ways of communicating with I/O devices and standard I/O interfaces. • To study the hierarchical memory structure.
Course Outcome:	On completion of the course, a student should be able to: • Able to distinguish between computer organization and computer architecture. • Able to store data in different ways in memory. • Able to Able to use different instructions in different circumstances. • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • Able to demonstrate different memory management policies. • Able to evaluate the performance of CPU, Memory, and I/O devices.
Syllabus	
Module I **[10] Hours**	**Module I** Basic structures of Computers: Functional units, operational concepts, Bus structures, Software, Computer Architecture vs Computer Organization. Quantitative measures of computer performance, Amdahl's law. Machine Instruction and Programs: Memory location and addresses, Big-endian and Little-endian representation. Instructions and instruction Sequencing, Addressing modes, Memory Operations, Basic Input/output operations, Subroutines, additional Instructions.
Module II **[8] Hours**	**Module II** Arithmetic: Addition and subtraction of signed Numbers, Design of Fast Adders, Multiplication of positive Numbers, Signed-operand multiplication,

	Fast multiplication, Integer Division, Floating- point Numbers, (IEEE754 s…) and operations.
Module III **[12] Hours**	**Module III** Basic Processing units: Fundamental concepts, execution of complete Instructions, Multi bus organization, Hardwired control, Micro programmed control, RISC vs CISC architecture. Memory System: Basic Concepts, Cache Memory, Cache memory mapping policies, Cache updating schemes, Concepts of Virtual memory. I/O Organization: Different modes of I/O, Programmed I/O, Memory mapped I/O, Interrupt initiated I/O, DMA and DMA controller.
Suggested Books:	1. Computer Organization: Carl Hamacher, Zvonkovranesic, Safwat Zaky,Mc Graw Hill,5th Ed 2. Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier, 4th Edition. 3. Computer Architecture and Organization: William Stallings, Pearson Education. 4. Computer Architecture and Organizations, Design principles and Application: B. Govinda Rajalu, Tata McGraw-Hill Publishing company Ltd.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	COMPUTER ORGANIZATION AND ARCHITECTURE LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basics of Digital Electronics
Course Objective:	The course objective is to: • Learn to create different digital circuits • Be familiarized with different simulating tools • Have a good understanding of internal working mechanism of CPU • Familiarize with different algorithms. • Be Exposed to different applications of low level computing
Course Outcome:	On completion of the course, a student should be able to: • Design and implement architecture of a computer. • Design different digital circuits.

Syllabus (Experiments Lists)

Experiment 1

(i) Identification of different components of a PC.
(ii) Assembling & disassembling of a PC.

Experiment 2

Simulate various architectures like accumulator-based, stack-based, and RISC using CPU Sim tool.

Experiment 3

Understanding the micro architecture of a program written in high-level language using a simulator like Escape or WinDLX or CPU Sim.

Experiment 4

Design of digital circuits (H/A, F/A, Decoder and Encoder) in VHDL using Active VHDL.

Experiment 5

Design of digital circuits (MUX, DEMUX and ALU) in VHDL using Active VHDL.

Experiment 6

Design of arithmetic circuits using Logisim tool.

Experiment 7

Design of control unit using CEDAR tool.

Experiment 8

(i) Write a C/C++ program to perform signed bit multiplication using Booth's algorithm.
(ii) Write a C/C++ program for IEEE-754 floating point representation and perform Addition/Subtraction.

Experiment 9

Simulate main memory to cache mapping techniques using Simple-Scalar or Gem5 or any other suitable simulator.

Experiment 10 Low level programming in C that emphasizes the power of low-level computing, as an aid to understand computing systems in depth:

(i) Pointers to functions and their applications.
(ii) Self-modifying code and applications.

Suggested Books:	1. Computer Organization: Carl Hamacher, Zvonkovranesic, Safwat Zaky,Mc Graw Hill,5th Ed

	2. Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier, 4th Edition. 3. Computer Architecture and Organization: William Stallings, Pearson Education. 4. Computer Architecture and Organizations, Design principles and Application: B. Govinda Rajalu, Tata McGraw-Hill Publishing company Ltd. 5.
Evaluation:	1. Lab Assessment: 50% (Continuous Assessment) 2. Mid Term: 20% 3. End Term: 30%

CSXXX	DATABASE MANAGEMENT SYSTEM (DBMS)
Credits: 4 (3-1-0)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objectives: • To Teach the basic database concepts, applications, data models, schemas and instances. • To familiarize Entity Relationship model for a database. • To Demonstrate the use of constraints and relational algebra operations. • To Describe the basics of SQL and construct queries using SQL. • To Emphasize the importance of normalization in databases. • To Demonstrate the basic concepts of transaction processing and concurrency control.
Course Outcome:	On completion of the course, a student should be able to: • Use the basic concepts of Database Systems in Database design. • Apply SQL queries to interact with Database 3. Design a Database using ER Modelling • Apply normalization on database design to eliminate anomalies. • Analyze database transactions and can control them by applying ACID properties.
Syllabus	
Module I **[8] Hours**	**Introduction and Conceptual Modelling** Basic Concepts: Data Modelling for a Database, Records and Files, File system vs DBMS, Three-Level Architecture for DBMS, Components of a DBMS. Data Models: Data Association, Data Models Classification: Object-Oriented and Object-Relational. Data Modelling using Entity-Relationship Model: Entity, Relationship, Attribute, Participation, Cardinality, Constraints in ER-Modelling, ER-Diagram, ER-Diagram to Relational- Table Conversion.
Module II **[10] Hours**	**Database Design Theory and Methodology** Functional Dependencies: Basic Concepts, Properties, and Classification. Armstrong's Axioms, Closure Set of Functional Dependencies, Identification of Additional Functional Dependencies, Equivalence between two Functional

	Dependencies, Identification of Keys, Minimal and Canonical Covers. Normalization: Database Anomalies, Needs of Normalization, Literatures related to Normalization, First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form, Fourth Normal Form, Fifth Normal Form. Properties of Decomposition: Lossless Join Property, Dependency Preserving Property, Identification of Good or Bad Decomposition.
Module III **[10] Hours**	**Relational Query Languages, Optimization and Design** SQL : Basics of DDL, DML, DCL, TCL. Select, Where, From, Having, Group by Clause. Database Constraints, Aggregation Functions, Join Operations, Views. Relational Algebra, Query processing and Optimization: Evaluation of Relational Algebra Expressions, Physical Design: Indexing, B-Tree, B^+ Tree.
Module IV **[12] Hours**	**Transaction Processing Concepts and Database Recovery** Transaction Processing: Transaction concepts, ACID Property, Schedule, Serializability. Concurrency Control Protocols: Needs of Concurrency Control, Lock based Protocol, Time Stamp based Protocol, Multiversion Time Stamp based Protocol, Optimistic Protocol (Validation based Protocol). Deadlock in Database: Prevention, Detection, Recovery. Database Recovery System: Types of Database Failure and Recovery, Recovery Technique: Log based Recovery: Deferred Database Modification, Immediate Database Modification.
Suggested Books:	1. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2005), Database System Concepts, 5th edition, McGraw-Hill, New Delhi, India. 3. Raghurama Krishnan, Johannes Gehrke , Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India. 4. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7th edition.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	DATABASE MANAGEMENT SYSTEM LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objective is to: • Learn to create and use a database • Be familiarized with a query language • Have hands on experience on DDL Commands • Have a good understanding of DML Commands and DCL commands • Familiarize advanced SQL queries. • Be Exposed to different applications
Course Outcome:	On completion of the course, a student should be able to: • Design and implement a database schema for a given problem-domain. • Populate and query a database. • Create and maintain tables using PL/SQL.

Syllabus (Experiments Lists)

Experiment 1
Student should decide on a case study and formulate the problem statement.

Experiment 2
Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.

Experiment 3 Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys) Note: Student is required to submit a document showing the database tables created from ER Model.

Experiment 4 Normalization -To remove the redundancies and anomalies in the above relational tables, Normalize up to Third Normal Form.

Experiment 5 Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables.

Experiment 6 Practicing DML commands- Insert, Select, Update, Delete

Experiment 7 Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc.

Experiment 8 Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Experiment 9 Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping.
Experiment 10 Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger.

Experiment 11 Procedures- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure.

Experiment 12 Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

Suggested Books:	1. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2005), Database System Concepts, 5th edition, McGraw-Hill, New Delhi, India.

	3. Raghurama Krishnan, Johannes Gehrke , Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India. 4. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7th edition.
Evaluation:	1. Lab Assessment: 50% 2. Mid Term: 20% 3. End Term: 30%

CS105	Theory Of Computation
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Discreet Mathematics
Course Objective:	• To understand the science behind computing by the introducing the mathematical foundations including automata design, study of formal language, grammar, algorithm, computability, and complexity.
Course Outcome:	On completion of the course, a student should be able to: • Students will be able to demonstrate the knowledge of basic mathematics of computation, i.e various automaton, and their expressive power and limitation, also able to relate it to their corresponding formal languages. • Students will come to know the capability of present computing models and learn undecidable problems. • Learn about problems that does not admit efficient algorithms, and study on the complexity theory.
Syllabus	
Module I **[13] Hours**	**Introduction:** Strings, languages, computational problems, finite representations, Chomsky hierarchy. **Finite Automata & Regular Languages:** Deterministic finite automata, applications of finite automata, regular languages, closure properties, Pumping lemma for regular languages, Myhill–Nerode theorem, State minimization algorithm. NFA, regular expressions, Regular Expressions and Regular languages, more closure properties of regular languages
Module II **[9] Hours**	**Context–free Languages and PDA:** Context–free Grammars and Languages: Parse trees, Applications of context free grammars, Ambiguity in grammars and languages. Pushdown Automata: Pushdown automation (PDA), the language of PDA, equivalence of PDA's and CFG's, Non-deterministic Pushdown Automata, Properties of Context – Free Languages: Normal forms of context free grammars, pumping lemma for context free languages, close properties of context free languages, Decision problems for CFLs.
Module III **[8] Hours**	**Turing Machine & Computability:** Introduction to Turing Machine: The Turing machine, Language acceptance by TMs, extensions to the basic Turing machine, restricted Turing Machines, Church-Turing hypothesis and its foundational implications, Codes for TMs. Recursively enumerable (R.E.) and recursive languages. Universal language and universal TM. Recursive

	and R.E. classes. Turing Machines and Computers. Undecidable problems of TMs. Rice's theorem. Undecidability of Post's correspondence problem (PCP), some simple applications of undecidability of PCP. **Complexity theory:** Complexity classes P, NP, Co-NP, NP-Complete and NP-Hard, some NP-complete problems, Polynomial time reduction, Cook-Levin Theorem.
Suggested Books:	1. J.E. Hopcroft, et.al. - Introduction to Automata Theory, Languages and Computation, 2nd Edn. Pearson Education, New Delhi 2001. 2. Michael Sipser, Introduction to the Theory of Computation, Books/Cole Thomson Learning, 2001. 3. J.C. Martin - Introduction to Languages and the Theory of Computation 4nd Edn, MH, New Delhi, 2011. 4. Dexter C Kozen, Automata and Computability, Springer, 1997 5. Harry R Lewis and Christos H Papadimitriou, Elements of the Theory of Computation, second edition, Prentice Hall, 1998.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	COMPUTER ORGANIZATION AND ARCHITECTURE
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Basic of Digital Electronics
Course Objective:	• To have a thorough understanding of the basic organization and architecture of a digital computer. • To study different ways of communicating with I/O devices and standard I/O interfaces. • To study the hierarchical memory structure.
Course Outcome:	On completion of the course, a student should be able to: • Able to distinguish between computer organization and computer architecture. • Able to store data in different ways in memory. • Able to Able to use different instructions in different circumstances. • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • Able to demonstrate different memory management policies. • Able to evaluate the performance of CPU, Memory, and I/O devices.
Syllabus	
Module I **[10] Hours**	**Module I** Basic structures of Computers: Functional units, operational concepts, Bus structures, Software, Computer Architecture vs Computer Organization. Quantitative measures of computer performance, Amdahl's law. Machine Instruction and Programs: Memory location and addresses, Big-endian and Little-endian representation. Instructions and instruction Sequencing, Addressing modes, Memory Operations, Basic Input/output operations, Subroutines, additional Instructions.
Module II **[8] Hours**	**Module II** Arithmetic: Addition and subtraction of signed Numbers, Design of Fast Adders, Multiplication of positive Numbers, Signed-operand multiplication,

	Fast multiplication, Integer Division, Floating- point Numbers, (IEEE754 s…) and operations.
Module III **[12] Hours**	**Module III** Basic Processing units: Fundamental concepts, execution of complete Instructions, Multi bus organization, Hardwired control, Micro programmed control, RISC vs CISC architecture. Memory System: Basic Concepts, Cache Memory, Cache memory mapping policies, Cache updating schemes, Concepts of Virtual memory. I/O Organization: Different modes of I/O, Programmed I/O, Memory mapped I/O, Interrupt initiated I/O, DMA and DMA controller.
Suggested Books:	1. Computer Organization: Carl Hamacher, Zvonkovranesic, Safwat Zaky,Mc Graw Hill,5th Ed 2. Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier, 4th Edition. 3. Computer Architecture and Organization: William Stallings, Pearson Education. 4. Computer Architecture and Organizations, Design principles and Application: B. Govinda Rajalu, Tata McGraw-Hill Publishing company Ltd.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	COMPUTER ORGANIZATION AND ARCHITECTURE LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basics of Digital Electronics
Course Objective:	The course objective is to: • Learn to create different digital circuits • Be familiarized with different simulating tools • Have a good understanding of internal working mechanism of CPU • Familiarize with different algorithms. • Be Exposed to different applications of low level computing
Course Outcome:	On completion of the course, a student should be able to: • Design and implement architecture of a computer. • Design different digital circuits.

Syllabus (Experiments Lists)

Experiment 1

(i) Identification of different components of a PC.
(ii) Assembling & disassembling of a PC.

Experiment 2

Simulate various architectures like accumulator-based, stack-based, and RISC using CPU Sim tool.

Experiment 3

Understanding the micro architecture of a program written in high-level language using a simulator like Escape or WinDLX or CPU Sim.

Experiment 4

Design of digital circuits (H/A, F/A, Decoder and Encoder) in VHDL using Active VHDL.

Experiment 5

Design of digital circuits (MUX, DEMUX and ALU) in VHDL using Active VHDL.

Experiment 6

Design of arithmetic circuits using Logisim tool.

Experiment 7

Design of control unit using CEDAR tool.

Experiment 8

(i) Write a C/C++ program to perform signed bit multiplication using Booth's algorithm.
(ii) Write a C/C++ program for IEEE-754 floating point representation and perform Addition/Subtraction.

Experiment 9

Simulate main memory to cache mapping techniques using Simple-Scalar or Gem5 or any other suitable simulator.

Experiment 10 Low level programming in C that emphasizes the power of low-level computing, as an aid to understand computing systems in depth:

(i) Pointers to functions and their applications.
(ii) Self-modifying code and applications.

Suggested Books:	1. Computer Organization: Carl Hamacher, Zvonkovranesic, Safwat Zaky,Mc Graw Hill,5th Ed

	2. Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier, 4th Edition. 3. Computer Architecture and Organization: William Stallings, Pearson Education. 4. Computer Architecture and Organizations, Design principles and Application: B. Govinda Rajalu, Tata McGraw-Hill Publishing company Ltd. 5.
Evaluation:	1. Lab Assessment: 50% (Continuous Assessment) 2. Mid Term: 20% 3. End Term: 30%

CSXXX	DATABASE MANAGEMENT SYSTEM (DBMS)
Credits: 4 (3-1-0)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objectives: • To Teach the basic database concepts, applications, data models, schemas and instances. • To familiarize Entity Relationship model for a database. • To Demonstrate the use of constraints and relational algebra operations. • To Describe the basics of SQL and construct queries using SQL. • To Emphasize the importance of normalization in databases. • To Demonstrate the basic concepts of transaction processing and concurrency control.
Course Outcome:	On completion of the course, a student should be able to: • Use the basic concepts of Database Systems in Database design. • Apply SQL queries to interact with Database 3. Design a Database using ER Modelling • Apply normalization on database design to eliminate anomalies. • Analyze database transactions and can control them by applying ACID properties.
Syllabus	
Module I **[8] Hours**	**Introduction and Conceptual Modelling** Basic Concepts: Data Modelling for a Database, Records and Files, File system vs DBMS, Three-Level Architecture for DBMS, Components of a DBMS. Data Models: Data Association, Data Models Classification: Object-Oriented and Object-Relational. Data Modelling using Entity-Relationship Model: Entity, Relationship, Attribute, Participation, Cardinality, Constraints in ER-Modelling, ER-Diagram, ER-Diagram to Relational- Table Conversion.
Module II **[10] Hours**	**Database Design Theory and Methodology** Functional Dependencies: Basic Concepts, Properties, and Classification. Armstrong's Axioms, Closure Set of Functional Dependencies, Identification of Additional Functional Dependencies, Equivalence between two Functional

	Dependencies, Identification of Keys, Minimal and Canonical Covers. Normalization: Database Anomalies, Needs of Normalization, Literatures related to Normalization, First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form, Fourth Normal Form, Fifth Normal Form. Properties of Decomposition: Lossless Join Property, Dependency Preserving Property, Identification of Good or Bad Decomposition.
Module III **[10] Hours**	**Relational Query Languages, Optimization and Design** SQL : Basics of DDL, DML, DCL, TCL. Select, Where, From, Having, Group by Clause. Database Constraints, Aggregation Functions, Join Operations, Views. Relational Algebra, Query processing and Optimization: Evaluation of Relational Algebra Expressions, Physical Design: Indexing, B-Tree, B^+ Tree.
Module IV **[12] Hours**	**Transaction Processing Concepts and Database Recovery** Transaction Processing: Transaction concepts, ACID Property, Schedule, Serializability. Concurrency Control Protocols: Needs of Concurrency Control, Lock based Protocol, Time Stamp based Protocol, Multiversion Time Stamp based Protocol, Optimistic Protocol (Validation based Protocol). Deadlock in Database: Prevention, Detection, Recovery. Database Recovery System: Types of Database Failure and Recovery, Recovery Technique: Log based Recovery: Deferred Database Modification, Immediate Database Modification.
Suggested Books:	1. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2005), Database System Concepts, 5th edition, McGraw-Hill, New Delhi, India. 3. Raghurama Krishnan, Johannes Gehrke , Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India. 4. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7th edition.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	DATABASE MANAGEMENT SYSTEM LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objective is to: • Learn to create and use a database • Be familiarized with a query language • Have hands on experience on DDL Commands • Have a good understanding of DML Commands and DCL commands • Familiarize advanced SQL queries. • Be Exposed to different applications
Course Outcome:	On completion of the course, a student should be able to: • Design and implement a database schema for a given problem-domain. • Populate and query a database. • Create and maintain tables using PL/SQL.

Syllabus (Experiments Lists)

Experiment 1
Student should decide on a case study and formulate the problem statement.

Experiment 2
Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.

Experiment 3 Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys) Note: Student is required to submit a document showing the database tables created from ER Model.

Experiment 4 Normalization -To remove the redundancies and anomalies in the above relational tables, Normalize up to Third Normal Form.

Experiment 5 Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables.

Experiment 6 Practicing DML commands- Insert, Select, Update, Delete

Experiment 7 Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc.

Experiment 8 Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Experiment 9 Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping.
Experiment 10 Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger.

Experiment 11 Procedures- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure.

Experiment 12 Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

Suggested Books:	1. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2005), Database System Concepts, 5th edition, McGraw-Hill, New Delhi, India.

	3. Raghurama Krishnan, Johannes Gehrke , Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India. 4. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7th edition.
Evaluation:	1. Lab Assessment: 50% 2. Mid Term: 20% 3. End Term: 30%

CS105	Theory Of Computation
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Discreet Mathematics
Course Objective:	• To understand the science behind computing by the introducing the mathematical foundations including automata design, study of formal language, grammar, algorithm, computability, and complexity.
Course Outcome:	On completion of the course, a student should be able to: • Students will be able to demonstrate the knowledge of basic mathematics of computation, i.e various automaton, and their expressive power and limitation, also able to relate it to their corresponding formal languages. • Students will come to know the capability of present computing models and learn undecidable problems. • Learn about problems that does not admit efficient algorithms, and study on the complexity theory.
Syllabus	
Module I **[13] Hours**	**Introduction:** Strings, languages, computational problems, finite representations, Chomsky hierarchy. **Finite Automata & Regular Languages:** Deterministic finite automata, applications of finite automata, regular languages, closure properties, Pumping lemma for regular languages, Myhill–Nerode theorem, State minimization algorithm. NFA, regular expressions, Regular Expressions and Regular languages, more closure properties of regular languages
Module II **[9] Hours**	**Context–free Languages and PDA:** Context–free Grammars and Languages: Parse trees, Applications of context free grammars, Ambiguity in grammars and languages. Pushdown Automata: Pushdown automation (PDA), the language of PDA, equivalence of PDA's and CFG's, Non-deterministic Pushdown Automata, Properties of Context – Free Languages: Normal forms of context free grammars, pumping lemma for context free languages, close properties of context free languages, Decision problems for CFLs.
Module III **[8] Hours**	**Turing Machine & Computability:** Introduction to Turing Machine: The Turing machine, Language acceptance by TMs, extensions to the basic Turing machine, restricted Turing Machines, Church-Turing hypothesis and its foundational implications, Codes for TMs. Recursively enumerable (R.E.) and recursive languages. Universal language and universal TM. Recursive

	and R.E. classes. Turing Machines and Computers. Undecidable problems of TMs. Rice's theorem. Undecidability of Post's correspondence problem (PCP), some simple applications of undecidability of PCP. **Complexity theory:** Complexity classes P, NP, Co-NP, NP-Complete and NP-Hard, some NP-complete problems, Polynomial time reduction, Cook-Levin Theorem.
Suggested Books:	1. J.E. Hopcroft, et.al. - Introduction to Automata Theory, Languages and Computation, 2nd Edn. Pearson Education, New Delhi 2001. 2. Michael Sipser, Introduction to the Theory of Computation, Books/Cole Thomson Learning, 2001. 3. J.C. Martin - Introduction to Languages and the Theory of Computation 4nd Edn, MH, New Delhi, 2011. 4. Dexter C Kozen, Automata and Computability, Springer, 1997 5. Harry R Lewis and Christos H Papadimitriou, Elements of the Theory of Computation, second edition, Prentice Hall, 1998.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	COMPUTER ORGANIZATION AND ARCHITECTURE
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Basic of Digital Electronics
Course Objective:	• To have a thorough understanding of the basic organization and architecture of a digital computer. • To study different ways of communicating with I/O devices and standard I/O interfaces. • To study the hierarchical memory structure.
Course Outcome:	On completion of the course, a student should be able to: • Able to distinguish between computer organization and computer architecture. • Able to store data in different ways in memory. • Able to Able to use different instructions in different circumstances. • Able to demonstrate the organization of a digital computer and describe the various principles and operations of different components of a digital computer. • Able to demonstrate different memory management policies. • Able to evaluate the performance of CPU, Memory, and I/O devices.
Syllabus	
Module I **[10] Hours**	**Module I** Basic structures of Computers: Functional units, operational concepts, Bus structures, Software, Computer Architecture vs Computer Organization. Quantitative measures of computer performance, Amdahl's law. Machine Instruction and Programs: Memory location and addresses, Big-endian and Little-endian representation. Instructions and instruction Sequencing, Addressing modes, Memory Operations, Basic Input/output operations, Subroutines, additional Instructions.
Module II **[8] Hours**	**Module II** Arithmetic: Addition and subtraction of signed Numbers, Design of Fast Adders, Multiplication of positive Numbers, Signed-operand multiplication,

	Fast multiplication, Integer Division, Floating- point Numbers, (IEEE754 s…) and operations.
Module III **[12] Hours**	**Module III** Basic Processing units: Fundamental concepts, execution of complete Instructions, Multi bus organization, Hardwired control, Micro programmed control, RISC vs CISC architecture. Memory System: Basic Concepts, Cache Memory, Cache memory mapping policies, Cache updating schemes, Concepts of Virtual memory. I/O Organization: Different modes of I/O, Programmed I/O, Memory mapped I/O, Interrupt initiated I/O, DMA and DMA controller.
Suggested Books:	1. Computer Organization: Carl Hamacher, Zvonkovranesic, Safwat Zaky,Mc Graw Hill,5th Ed 2. Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier, 4th Edition. 3. Computer Architecture and Organization: William Stallings, Pearson Education. 4. Computer Architecture and Organizations, Design principles and Application: B. Govinda Rajalu, Tata McGraw-Hill Publishing company Ltd.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	COMPUTER ORGANIZATION AND ARCHITECTURE LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basics of Digital Electronics
Course Objective:	The course objective is to: • Learn to create different digital circuits • Be familiarized with different simulating tools • Have a good understanding of internal working mechanism of CPU • Familiarize with different algorithms. • Be Exposed to different applications of low level computing
Course Outcome:	On completion of the course, a student should be able to: • Design and implement architecture of a computer. • Design different digital circuits.

Syllabus (Experiments Lists)

Experiment 1

(i) Identification of different components of a PC.
(ii) Assembling & disassembling of a PC.

Experiment 2

Simulate various architectures like accumulator-based, stack-based, and RISC using CPU Sim tool.

Experiment 3

Understanding the micro architecture of a program written in high-level language using a simulator like Escape or WinDLX or CPU Sim.

Experiment 4

Design of digital circuits (H/A, F/A, Decoder and Encoder) in VHDL using Active VHDL.

Experiment 5

Design of digital circuits (MUX, DEMUX and ALU) in VHDL using Active VHDL.

Experiment 6

Design of arithmetic circuits using Logisim tool.

Experiment 7

Design of control unit using CEDAR tool.

Experiment 8

(i) Write a C/C++ program to perform signed bit multiplication using Booth's algorithm.
(ii) Write a C/C++ program for IEEE-754 floating point representation and perform Addition/Subtraction.

Experiment 9

Simulate main memory to cache mapping techniques using Simple-Scalar or Gem5 or any other suitable simulator.

Experiment 10 Low level programming in C that emphasizes the power of low-level computing, as an aid to understand computing systems in depth:

(i) Pointers to functions and their applications.
(ii) Self-modifying code and applications.

Suggested Books:	1. Computer Organization: Carl Hamacher, Zvonkovranesic, Safwat Zaky,Mc Graw Hill,5th Ed

	2. Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier, 4th Edition. 3. Computer Architecture and Organization: William Stallings, Pearson Education. 4. Computer Architecture and Organizations, Design principles and Application: B. Govinda Rajalu, Tata McGraw-Hill Publishing company Ltd. 5.
Evaluation:	1. Lab Assessment: 50% (Continuous Assessment) 2. Mid Term: 20% 3. End Term: 30%

CSXXX	DATABASE MANAGEMENT SYSTEM (DBMS)
Credits: 4 (3-1-0)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objectives: • To Teach the basic database concepts, applications, data models, schemas and instances. • To familiarize Entity Relationship model for a database. • To Demonstrate the use of constraints and relational algebra operations. • To Describe the basics of SQL and construct queries using SQL. • To Emphasize the importance of normalization in databases. • To Demonstrate the basic concepts of transaction processing and concurrency control.
Course Outcome:	On completion of the course, a student should be able to: • Use the basic concepts of Database Systems in Database design. • Apply SQL queries to interact with Database 3. Design a Database using ER Modelling • Apply normalization on database design to eliminate anomalies. • Analyze database transactions and can control them by applying ACID properties.
Syllabus	
Module I **[8] Hours**	**Introduction and Conceptual Modelling** Basic Concepts: Data Modelling for a Database, Records and Files, File system vs DBMS, Three-Level Architecture for DBMS, Components of a DBMS. Data Models: Data Association, Data Models Classification: Object-Oriented and Object-Relational. Data Modelling using Entity-Relationship Model: Entity, Relationship, Attribute, Participation, Cardinality, Constraints in ER-Modelling, ER-Diagram, ER-Diagram to Relational- Table Conversion.
Module II **[10] Hours**	**Database Design Theory and Methodology** Functional Dependencies: Basic Concepts, Properties, and Classification. Armstrong's Axioms, Closure Set of Functional Dependencies, Identification of Additional Functional Dependencies, Equivalence between two Functional

	Dependencies, Identification of Keys, Minimal and Canonical Covers. Normalization: Database Anomalies, Needs of Normalization, Literatures related to Normalization, First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form, Fourth Normal Form, Fifth Normal Form. Properties of Decomposition: Lossless Join Property, Dependency Preserving Property, Identification of Good or Bad Decomposition.
Module III **[10] Hours**	**Relational Query Languages, Optimization and Design** SQL : Basics of DDL, DML, DCL, TCL. Select, Where, From, Having, Group by Clause. Database Constraints, Aggregation Functions, Join Operations, Views. Relational Algebra, Query processing and Optimization: Evaluation of Relational Algebra Expressions, Physical Design: Indexing, B-Tree, B^+ Tree.
Module IV **[12] Hours**	**Transaction Processing Concepts and Database Recovery** Transaction Processing: Transaction concepts, ACID Property, Schedule, Serializability. Concurrency Control Protocols: Needs of Concurrency Control, Lock based Protocol, Time Stamp based Protocol, Multiversion Time Stamp based Protocol, Optimistic Protocol (Validation based Protocol). Deadlock in Database: Prevention, Detection, Recovery. Database Recovery System: Types of Database Failure and Recovery, Recovery Technique: Log based Recovery: Deferred Database Modification, Immediate Database Modification.
Suggested Books:	1. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2005), Database System Concepts, 5th edition, McGraw-Hill, New Delhi, India. 3. Raghurama Krishnan, Johannes Gehrke , Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India. 4. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7th edition.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

CSXXX	DATABASE MANAGEMENT SYSTEM LAB
Credits: 1 (0-0-2)	
Pre-requisite for this course:	• Basic of Data Structures and Programming
Course Objective:	The course objective is to: • Learn to create and use a database • Be familiarized with a query language • Have hands on experience on DDL Commands • Have a good understanding of DML Commands and DCL commands • Familiarize advanced SQL queries. • Be Exposed to different applications
Course Outcome:	On completion of the course, a student should be able to: • Design and implement a database schema for a given problem-domain. • Populate and query a database. • Create and maintain tables using PL/SQL.

Syllabus (Experiments Lists)

Experiment 1
Student should decide on a case study and formulate the problem statement.

Experiment 2
Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.

Experiment 3 Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys) Note: Student is required to submit a document showing the database tables created from ER Model.

Experiment 4 Normalization -To remove the redundancies and anomalies in the above relational tables, Normalize up to Third Normal Form.

Experiment 5 Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables.

Experiment 6 Practicing DML commands- Insert, Select, Update, Delete

Experiment 7 Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc.

Experiment 8 Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Experiment 9 Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping.

Experiment 10 Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger.

Experiment 11 Procedures- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure.

Experiment 12 Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

Suggested Books:	1. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2005), Database System Concepts, 5th edition, McGraw-Hill, New Delhi, India.

	3. Raghurama Krishnan, Johannes Gehrke , Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India. 4. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7th edition.
Evaluation:	1. Lab Assessment: 50% 2. Mid Term: 20% 3. End Term: 30%

CS105	Theory Of Computation
Credits: 3 (3-0-0)	
Pre-requisite for this course:	• Discreet Mathematics
Course Objective:	• To understand the science behind computing by the introducing the mathematical foundations including automata design, study of formal language, grammar, algorithm, computability, and complexity.
Course Outcome:	On completion of the course, a student should be able to: • Students will be able to demonstrate the knowledge of basic mathematics of computation, i.e various automaton, and their expressive power and limitation, also able to relate it to their corresponding formal languages. • Students will come to know the capability of present computing models and learn undecidable problems. • Learn about problems that does not admit efficient algorithms, and study on the complexity theory.
Syllabus	
Module I **[13] Hours**	**Introduction:** Strings, languages, computational problems, finite representations, Chomsky hierarchy. **Finite Automata & Regular Languages:** Deterministic finite automata, applications of finite automata, regular languages, closure properties, Pumping lemma for regular languages, Myhill–Nerode theorem, State minimization algorithm. NFA, regular expressions, Regular Expressions and Regular languages, more closure properties of regular languages
Module II **[9] Hours**	**Context–free Languages and PDA:** Context–free Grammars and Languages: Parse trees, Applications of context free grammars, Ambiguity in grammars and languages. Pushdown Automata: Pushdown automation (PDA), the language of PDA, equivalence of PDA's and CFG's, Non-deterministic Pushdown Automata, Properties of Context – Free Languages: Normal forms of context free grammars, pumping lemma for context free languages, close properties of context free languages, Decision problems for CFLs.
Module III **[8] Hours**	**Turing Machine & Computability:** Introduction to Turing Machine: The Turing machine, Language acceptance by TMs, extensions to the basic Turing machine, restricted Turing Machines, Church-Turing hypothesis and its foundational implications, Codes for TMs. Recursively enumerable (R.E.) and recursive languages. Universal language and universal TM. Recursive

	and R.E. classes. Turing Machines and Computers. Undecidable problems of TMs. Rice's theorem. Undecidability of Post's correspondence problem (PCP), some simple applications of undecidability of PCP. **Complexity theory:** Complexity classes P, NP, Co-NP, NP-Complete and NP-Hard, some NP-complete problems, Polynomial time reduction, Cook-Levin Theorem.
Suggested Books:	1. J.E. Hopcroft, et.al. - Introduction to Automata Theory, Languages and Computation, 2nd Edn. Pearson Education, New Delhi 2001. 2. Michael Sipser, Introduction to the Theory of Computation, Books/Cole Thomson Learning, 2001. 3. J.C. Martin - Introduction to Languages and the Theory of Computation 4nd Edn, MH, New Delhi, 2011. 4. Dexter C Kozen, Automata and Computability, Springer, 1997 5. Harry R Lewis and Christos H Papadimitriou, Elements of the Theory of Computation, second edition, Prentice Hall, 1998.
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%