Dept. of Computer Science and Engineering

Jahangirnagar University

Syllabus for B.Sc. (Hons.) in Computer Science and Engineering

(Effective from 2018-19)

Detail Syllabus of First Year Second Semester

Course code	**:**	CSE 150	**Credit**	**:**	1.0
Title	**:**	Viva-Voce	**Prerequisite**	**:**	None
Type	**:**	*Viva-Voce*	**Contact hours**	**:**	-

Rationale
Viva-Voce is used to measure and evaluate the students through oral examination on their previous taught/learned courses so that students have ability to face viva-board confidently in their professional life.

Course Objectives
Measure and evaluate the students through oral examination on their previous taught/learned courses

Students Learning Outcomes
After successful completion of this course, students should be able to: Expose their views orally in different situations on diverse fields of Computer Science and Engineering

#	**Title and Descriptions**
	The viva-voce will be held on all the courses of first year second semester.

References
The reading materials provided by the Course Teachers for all the courses of first year second semester

Course code	**:**	MATH 151	**Credit**	**:**	3.0
Title	**:**	Mathematics II (Integral Calculus, Differential Equations and Series Solution)	**Prerequisite**	**:**	N/A
Type	**:**	*Theory*	**Contact hours**	**:**	39

Rationale

Integral calculus involves the area between the graph of a function and the horizontal axis; Differential equation models describe a wide range of complex problems in biology, engineering, physical sciences, economics and finance and the power series method is used to seek a power series solution to certain differential equations. This course is designed for the students to give the concepts of Integral calculus, Differential Calculus and series solution so that they can solve real world problems.

Course Objectives

- To know the relation between the area and anti-derivatives and how to calculate an anti-derivative.
- To learn the classifications of differential equations and many method and techniques to solve real world problems.
- To understand what series solution of differential equation is and in what case it exits and also its method of solution

Students Learning Outcomes

After successful completion of this course, students should be able to:

- Learn the real understanding of the connection between area and integration.
- Find the techniques to derive different integration by various methods
- Classify differential equations by order, linearity, and homogeneity.
- Solve exact differential equations, first order linear and nonlinear differential equations and higher degree differential equations.

#	Title and Descriptions
1	**Integral Calculus I** Definitions of integration; Integration by the method of substitutions; Integration by parts; Standard integrals; Integration by the method of successive reduction; Definite integrals and its properties and use in summing series; Walli’s formula, Improper integrals, Beta function and Gamma function;
2	**Integral Calculus II** Area under a plane curve in Cartesian and polar co-ordinates; Area of the region enclosed by two curves in Cartesian and polar co-ordinates; Trapezoidal rule, Simpson’s rule. Arc lengths of curves in Cartesian and polar co-ordinates, parametric and pedal equations; Intrinsic equation; Volume of solids of revolution; Volume of hollow solids of revolution by shell method. Area of surface of revolution; Jacobian, multiple integrals and their application.
3	**Ordinary Differential Equation I** Degree and order of ordinary differential equations; Formation of differential equations; Solution of first order differential equations by various methods; Solution of first order but higher degree ordinary differential equations; Solution of general linear equations of second and higher orders with constant coefficients;
4	**Ordinary Differential Equation II**

	Solution of homogeneous linear equations and its applications; Solution of differential equations of higher order when dependent and independent variables are absent; Solution of differential equation by the method based on factorization of operators.
5	**Partial Differential Equations I** Four rules for solving simultaneous equations of the form; Lagrange's method of solving PDE of order one; Integral surfaces passing through a given curve; Nonlinear PDE of order one (complete, particular, singular and general integrals): standard forms $f(p,q) = 0$, $z = px + qy + f(p,q)$, $f(p,q,z) = 0$, fÂ1(x,p) = f2(y, q);
6	**Partial Differential Equations II** Charpit's method; Second order PDE: its nomenclature and classifications to canonical (standard)- parabolic, elliptic, hyperbolic; Solution by separation of variables. Linear PDE with constant coefficients
7	**Series Solution** Solution of differential equations in series by the method of Frobenius; Bessel's functions, Legendre's polynomials and their properties.

Recommended Books				
1.	Integral calculus	B.C. Das	4th	
2.	Integral calculus.	A. Matin		
3.	Differential Equations.	B.O. Sharma		
4.	Ordinary and Partial differential equations	M.D. Raisingha mia		

Course code	**:**	CSE 153	**Credit**	**:**	3.0
Title	**:**	Discrete Mathematics	**Prerequisite**	**:**	N/A
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Discrete mathematics is the study of mathematics that underpins computer science, with a focus on discrete structures, for example, graphs, trees and networks. It discusses languages used in mathematical reasoning, basic concepts, and their properties and relationships among them. Discrete math will help to understand set theory, probability, and combinations which will allow to analyze algorithms. It provides excellent modelling tools for analysing real-world phenomena that varies in one state or another and is a vital tool used in a wide range of applications.

Course Objectives

- Simplify and evaluate basic logic statements.
- Express a logic sentence in terms of predicates, quantifiers, and logical connectives.
- Apply the operations of sets, rules of inference, tests for validity, and methods of proof.
- Solve counting problems.
- Describe binary relations between two sets.
- Define graphs, memory representation of graphs, binary trees, memory representation of binary trees.

Students Learning Outcomes
After successful completion of this course, students should be able to: • Express a logic sentence in terms of predicates, quantifiers, and logical connectives. • Apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction. • Use tree and graph algorithms to solve problems.

#	Title and Descriptions
1	**Sets and Functions** Sets, Set Operations, Functions, Sequences and Summations, Cardinality of Sets, Matrices. Algorithms, the Growth of Functions, Complexity of Algorithms.
2	**Relations** Relation, composition relation, Pictorial representation of relations, Properties of relations, Closure properties.
3	**Logic and Proofs** Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.
4	**Number Theory and Cryptography** Divisibility and Modular Arithmetic, Integer Representations and Algorithms, Primes and Greatest Common Divisors, Solving Congruence, Applications of Congruence, Cryptography.
5	**Counting** The Basics of Counting, the Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Generating Permutations and Combinations.
6	**Graphs** Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.
7	**Trees** Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

Recommended Books					
1.	Discrete mathematics and its applications	Rosen, K.	7th	McGraw-Hill	2012
2.	Schaum's outline: theory and problems of discrete mathematics	Lipschutz, S., & Lipson, M.	2nd	McGraw-Hill	197
3.	An introduction to discrete mathematics Harcourt Brace Jovanovich.	Roman, S.	2nd	San Diego	1989

Course code	**:**	CSE 155	**Credit**	**:**	3.0
Title	**:**	Data Structures	**Prerequisite**	**:**	C / C++ / Java
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

The course covers analysis and design of fundamental data structures and engages learners to use data structures as tools to algorithmically design efficient computer programs that will cope with the complexity of actual applications. The course focuses on basic and essential topics in data structures, including array-based lists, linked lists, stack and queues, hash tables, recursion, binary trees, heaps, sorting algorithms, and graphs.

Course Objectives

- To know the fundamental data structures of how computer store data in memory
- To use data structures as tools to algorithmically design efficient computer programs that will cope with the complexity of actual applications.
- To focus on basic and essential topics in data structures, including array-based lists, linked lists, stack and queues, hash tables, recursion, binary trees, heaps, sorting algorithms, and graphs.

Students Learning Outcomes

After successful completion of this course, students should be able to:

- Explain the need for efficiency in data structure.
- Apply methods to analyze running time of essential data structures and estimate efficiency of the procedures and implementations.
- Understand and apply the concept of abstract data type to represent and implement heterogeneous data structures.
- Write programs using array-based lists, Queues, linked lists etc.
- Demonstrate skills in tracing, analyzing, and designing recursive algorithms and recursive methods.
- Write programs using trees and graphs.

#	**Title and Descriptions**
1	**Introduction and String Processing** Basic terminology, Mathematical notation and functions, Complexity of algorithms. Storing techniques, Operations, word processing, and Pattern matching algorithm
2	**Arrays, Records and Pointers** Pointers, Structures, dynamic memory allocation and Abstract Data Type, Sorting and Searching Algorithms
3	**Linked Lists** Memory representation, Implementation and its application
4	**Stack, Queue and Recursion** Stack Implementation and its Application, Queue Implementation and its Application, Iterative Solution and Recursive Solution design

5	**Tree-01** Basic Tree Concepts, Tree Traversals, Binary Trees and their applications, Binary Search Tree: Insert, Delete, Search and Traversal Algorithms
6	**Tree-02** AVL Tree, B trees, Binary Heap and Priority queue, Spanning Tree, MST, General trees
7	**Graphs** Terminology, Graph representation, Graph traversal techniques, Shortest Path Problem, Hashing: Methods, Hashed Search

Recommended Books					
1.	Data Structures	Seymour Lipschutz	1st	McGraw Hill	2014
2.	Data Structures and Algorithms Made Easy	Narasimha Karumanchi	5th	Career Monk Plublications	2016
3.	Data Structures and Algorithms in C++	Adam Drozdek	4th	Cengage Learning	2012

Course code	**:**	CSE 156	**Credit**	**:**	1
Title	**:**	Data Structures Laboratory	**Prerequisite**	**:**	C / C++ / Java
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

In order to develop efficient software systems, it is essential that efficient algorithms and appropriate data structures are used. This course will help the students to develop efficient data structures and algorithms in a systematic manner.

Lab Objectives

- To help students to find problem definition
- To help students to find algorithm design
- To develop computer program based on theory course CSE-155 either in C or C++ or Java language

Lab Outcome

After successful completion of this course, students should be able to:

- Understand and apply basic data structures for storage and retrieval of ordered and unordered data.
- Implement and characterize algorithms for creation and manipulation of data structures like stacks, queues, linked list, etc.
- Interpret and apply appropriate data structures for implementing problem solving algorithms such as searching, insertion, deletion, traversing mechanism, etc., on various data structures.

- Compute and characterize the efficiency of data structures for complex problem-solving algorithms; perform and demonstrate this knowledge and write report for realistic problem solving.

#	Title and Descriptions
1	Design, Develop and Implement a Program in C for the following operations on Strings a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP) b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR Support the program with functions for each of the above operations. Don't use Built-in functions.
2	Design, Develop and Implement a menu driven Program in C for the following Array operations a. Creating an Array of N Integer Elements b. Display of Array Elements with Suitable Headings c. Inserting an Element (ELEM) at a given valid Position (POS) d. Deleting an Element at a given valid Position(POS) e. Exit. Support the program with functions for each of the above operations
3	Design, Develop and Implement a menu driven Program in C for the following Array operations a. Creating an Array of N Integer Elements b. Sort the elements using Bubble Sort Algorithm b. Search an item using Linear Search Algorithm c. Search an item using Binary Search Algorithm d. Exit. Support the program with functions for each of the above operations
4	Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, %(Remainder), ^ (Power) and alphanumeric operands.
5	Design, Develop and Implement a Program in C for the following Stack Applications a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ b. Solving Tower of Hanoi problem with n disks
6	Design, Develop and Implement a Program in C for the following Recursion Applications a. Calculate the factorial of n b. Display the Fibonacci sequence of n numbers c. $Q(J,K)=\begin{cases}5 & if\ J<K \\ Q(J-K,K+2)+J & if\ J\geq K\end{cases}$ d. Exit. Support the program with functions for each of the above operations.
7	Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE

	c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit Support the program with appropriate functions for each of the above operations
8	Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue f. Exit
9	Design, Develop and Implement a menu driven Program in C for the following operations on Complete Binary Search Tree of Integers a. Create a Heap of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Sort the list using Heap Sort Algorithm. c. Exit
10	Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities a. Create a Weighted Graph of N cities using Adjacency Matrix. b. Print the shortest path from a weighted graph using Warshall’s Algorithm.

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
Intel® Pentium® 4 Processor 1.60 GHz, 256K Cache, 400 MHz FSB / updated computers	Code Blocks / Quincy / Turbo C / Net beans / Eclipse

Course code	:	CSE 157	**Credit**	:	3.0
Title	:	Electronic Devices and Circuits-I	**Prerequisite**	:	None
Type	:	Theory	**Contact hours**	:	39

Rationale

This course is designed to understand the construction, theory and operation of the basic electronic devices such as PN junction diodes, Bipolar transistors and Field Effect Transistors and Optical devices.

Course Objectives

- To understand the construction, theory and operation of the basic electronic devices such as PN junction diodes, Bipolar transistors and Field Effect Transistors and Optical devices.

Students Learning Outcomes
After completion of the course, a student is able to construct and analyze the basic electronic machines and appliances.

#	Descriptions
1	**Crystal Structure of Solid** Semiconductor materials, Types of solids, Space lattices, Atomic bonding, Imperfections and impurities in solids, Growth of semiconductor materials
2	**Semiconductor in Equilibrium** Charge carriers in semiconductors, Dopant atoms and energy levels, Extrinsic semiconductor, Statistics of donors and acceptors, Charge neutrality, Position of Fermi energy level
3	**Carrier Transport Phenomena** Carrier drift, Carrier diffusion, Graded impurity diffusion, Hall effect - qualitative
4	**pn Junction Diode** Basic structure of pn junction, Zero applied bias, Reverse applied bias, pn junction current, Small-signal model of the pn junction, Generation-recombination currents (qualitative), Small-signal model of the pn junction, Junction breakdown, Tunnel diode
5	**Bipolar Transistor** Bipolar transistor action, Minority carrier distribution, Low frequency common-base current gain, Non-ideal effects, Base width modulation, Breakdown voltage, Equivalent circuit models, Frequency limitations
6	**Fundamentals of the MOSFET** The Two-terminal MOS structure, Energy-band diagrams, Depletion layer thickness, Work function differences, Surface Charge Density, Flat-band voltage, Threshold voltage, Charge distribution, Capacitance-voltage characteristics, Basic MOSFET characteristics, Frequency limitations, CMOS technology (qualitative)
7	**Optical Devices** Optical absorption, Solar cells, Photo detectors, Photoluminescence and electroluminescence, Light emitting diodes, Laser diode etc.

Recommended Books					
1.	Electronic Devices and Circuit Theory	R.L. Boylestad, L. Nashelshy		PHI	1999
2.	Solid State Electronic Devices	B.G. Streetman, S. Banerjee		Prentice Hall	2000
4.	Physical Properties of Semiconductors	C.M. Wolfe, N. Holonyak Jr., G.E. Stillman,		Prentice-Hall	1989
5.	Semiconductor Physics and Devices	Donald A Neaman,	3rd	Tata Mc GrawHill	2007

Course code	**:**	CSE 158	**Credit**	**:**	1
Title	**:**	Electronic Devices and Circuits-I Laboratory	**Prerequisite**	**:**	CSE 157
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

Using this laboratory course students will understand the characteristics of basic electronic devices and their usage in the common circuit applications.

Lab Objectives

- To understand the characteristics of basic electronic devices and their usage in the common circuit applications.

Lab Outcome

After successful completion of this laboratory course, students should be able to:

- Recognize the ICs containing the common electronic devices and understand their pin configuration.
- Apply the devices in various daily applications.

Lab Course Description

#	**Title**
1	Study of the characteristics of pn junction diode.
2	Construct and study of rectifier circuits using pn junction diode.
3	Study the behavior of Zener diode.
4	Study the common base configuration using Bipolar Junction Transistor (BJT) and its usage.
5	Study the common emitter configuration using Bipolar Junction Transistor (BJT) and its usage.
6	Study the common collector configuration using Bipolar Junction Transistor (BJT) and its usage.
7	Using BJT in electronic circuit as a switching component.
8	Study the characteristics of Field Effect Transistor (FET) and compare it with BJT.
9	Transmission of digital signal through fiber optic channel.

Hardware and Software Requirements

H/W Requirements	***S/W Requirements***
ICs, Bread boards, DC supply generator, Oscilloscope, Signal generators.	No special S/W is required.

Course code	**:**	CSE 159	**Credit**	**:**	3.0
Title	**:**	Object Oriented Programming (C++)	**Prerequisite**	**:**	CSE 105
Type	**:**	*Theory*	**Contact hours**	**:**	39

Rationale
Object Oriented Programming (OOP) aims to implement real world entities like inheritance, data hiding, polymorphism etc. in programming. The main aim of OOP is to bind together the data and the functions that operates on them so that no other part of code can access this data except that function. This unit concentrates on fundamental concepts of OOP which are essential for software engineering.

Course Objectives
The goal of this course is to provide essential knowledge of OOP and implement the concepts using C++ so that the students can draw an analogy between the theory and real world. The course will assume a good background knowledge in structured programming language like C.

Students Learning Outcomes
After completing this course students will be able to- • Codes basic programs(e.g. print message on console, relational operations, loops, arrays) in C++ programming language • Do different operations (e.g. represent, declare, use, manipulate, constructor, destructor, life cycle) objects and classes. • Program the object-oriented programming concepts (e.g. encapsulation, inheritance, polymorphism, interfaces, abstract classes and abstract methods). • Implement operator overloading functions. • Understand and handles exceptions • Uses generic classes and methods.

#	**Topics and Descriptions**
1.	**Overview of C++ Basic Syntax** C++ Syntax and Semantics, Operators, Data Types, Expressions and Output in C++, Program input, Conditions, Logical Expressions, Switch, Loops.
2.	**Arrays, Strings, Dynamic Memory Allocation and Functions** Declarations, Initialization, Accessing one of Multi-dimensional Array, Basic String Operations(find, insert, delete, replace), Functions, Recursion, Inline Functions,
3.	**Classes and Objects** Attributes, Methods, Access Modifier, Object, Constructor, Destructor, Copy Constructor, Memory Allocations.
4.	**OOP Basics**

	Encapsulation, Polymorphism, Overloading (Function and Operator), Pointer of Classes.
5.	**OOP Advanced** Inheritances, Abstract Class, Interfaces, Function Overriding, Associations, Aggregations, Reflection and Runtime Type Information.
6.	**Design Patterns** Overview, Creational(Abstract Factory, Builder, Factory, Lazy Initialization, Object Pool, Singleton), Structural(Adaptor, Bridge, Composite, Flyweight), Behavioral(Chain of Responsibility, Command, Iterator, Observer, Strategy)
7.	**Collection Frameworks, Generics, Exceptions, Files and Streams** C++ STL Containers, C++'s STL <algorithm> Library, Exception Handling, Files (opening and closing a files, ofstream, ifstream, fstream)

Recommended Books

1	Object-Oriented Programming in C++	Robert Lafore	4th	2008
2	C++ How to Program, 10th Edition	Paul J. Deitel	10th	2017
3	C++: The Complete Reference	Herbert Schildt	4th	2003
4	OOP with C++	Balaguruswamy	1st	2009

Course code	**:**	CSE 160	**Credit**	**:**	1
Title	**:**	Object Oriented Programming (C++) Laboratory	**Prerequisite**	**:**	*C programming*
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

OOP offers several benefits to the program designer and the user. Object-orientation contributes to the solutions of many problem associated with the development and quality of software products. The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost. This course is designed for the students to get the opportunity to learn about OOP paradigm.

Lab Objectives

- Identify and practice the object-oriented programming concepts and techniques,
- Understand and practice fundamentals of object-oriented programming in c++ classes, invoking methods and functions.
- Practice the use of C++ classes and class libraries, arrays, vectors, inheritance and file I/O stream concepts.

Lab Outcome
After successful completion of this laboratory course, students should be able to: • Create simple programs using classes and objects in C++. • Implement Object Oriented Programming Concepts in C++. • Develop applications using stream I/O and file I/O, templates and exceptional handling concepts.

Lab Course Description

#	**Title**
1	Practice Simple C++ Programs to Implement Various Control Structures. a. If statement b. Switch case statement and do while loop c. For loop d. While loop
2	Programs to Understand Structure & Unions.
3	Programs to Understand Pointer Arithmetic.
4	Programs to Understand Functions & Recursion.
5	Introduction to Classes and Objects, Access Specifiers, Constructors and Destructors
6	Constructor Overloading and Copy Constructors, Shallow Copy/ Deep Copy and Working with arrays
7	Friend Functions and Friend Classes, Operator Overloading and Inheritance
8	Multi-level and Multiple Inheritance, Function Overloading and Function Overriding
9	Polymorphism and Relationship
10	Function and Class Templates and Exception Handling

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
Desktop PC in Windows/Linux/Mac OS	g++ compiler IDE (Code Block, geany, or any)

Course code	**:**	CSE 162	**Credit**	**:**	1
Title	**:**	Technical Writing and Presentation Laboratory	**Prerequisite**	**:**	N/A
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale
This is a specialized technical writing lab course that will let students apply principles and techniques of writing and presenting technical material to subject matter from their major field of study.

Lab Objectives

- To help students to write easier and smoother, resulting in more effective technical documents.
- To present oneself masterpiece confidently.

Lab Outcome

After successful completion of this laboratory course, students should be able to:

- Analyze their audience and tailor the content to their specific needs
- Use best practice in structuring their document and choose words that support their documents
- Assess the best places to use graphics, and choose the right image to support their content
- Design and structure a document by analyzing the readership and selecting the right information
- Write clearly and in the correct style for their readers and use correct language and grammar
- Use layout, typography and illustrations to help get your message across
- Edit one’s draft for maximum impact

Lab Course Description

#	Title	Contact Hours
1	Technical Report Writing – the Basics	3
2	Technical Report Objectives – Primary, Secondary, Tertiary	3
3	Technical Report Strategy – Structuring the Framework	3
4	Technical Report Storyboarding – Enjoyable Reading	3
5	Technical Report Reverse Engineering – Reinforcing the Architect	3
6	Technical Report Writing – Preparing for the Report	3
7	Technical Report Writing – Generating the Report	3
8	Technical Report Writing – Completing the Report	3
9	Presenting Your Report and Proposal with Confidence	2

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
Desktop PC in Windows/Linux/Mac OS	Office Management Software, laTex