6.2 Course outline

Semester-wise course outline for the entire program

First Year				
	Couse Code	**Title**	**Credit**	**Remarks**
First Semester	CSE 100	Viva-Voce	1	**Theory : 15 Cr.** **Laboratory : 3 Cr.** **Viva-Voce : 1 Cr.**
	MATH 101	Mathematics I (Matrix, Differential Calculus and Coordinate Geometry)	3	
	ENG 103	Communicative English	3	
	CSE 105	Structured Programming	3	
	CSE 106	Structured Programming Laboratory	1	
	CSE 107	Electrical Circuits	3	
	CSE 108	Electrical Circuits Laboratory	1	
	PHY 109	Physics	3	
	URP 112	Computer Aided Engineering Drawing Laboratory	1	
			Total Credit	**19**
Second Semester	**Couse Code**	**Title**	**Credit**	**Remarks**
	CSE 150	Viva-Voce	1	**Theory : 15 Cr.** **Laboratory : 4 Cr.** **Viva-Voce : 1 Cr.**
	MATH 151	Mathematics II (Integral Calculus, Differential Equations and Series Solution)	3	
	CSE 153	Discrete Mathematics	3	
	CSE 155	Data Structures	3	
	CSE 156	Data Structures Laboratory	1	
	CSE 157	Electronic Devices and Circuits-I	3	
	CSE 158	Electronic Devices and Circuits-I Laboratory	1	
	CSE 159	Object Oriented Programming (C++)	3	
	CSE 160	Object Oriented Programming (C++) Laboratory	1	
	CSE 162	Technical Writing and Presentation Laboratory	1	
			Total Credit	**20**

Second Year				
	Couse Code	**Title**	**Credit**	**Remarks**
First Semester	CSE 200	Viva-Voce	1	**Theory : 15 Cr.** **Laboratory : 4 Cr.** **Viva-Voce : 1 Cr.**
	MATH 201	Mathematics –III (Vector, Complex Variable, Fourier Analysis and Laplace Transformation)	3	
	CSE 203	Computer Ethics and Cyber Law	3	
	CSE 205	Numerical Methods	3	
	CSE 206	Numerical Methods Laboratory	1	
	CSE 207	Electronic Devices and Circuits-II	3	
	CSE 208	Electronic Devices and Circuits-II Laboratory	1	
	CSE 209	Algorithms-I	3	
	CSE 210	Algorithms-I Laboratory	1	
	CSE 212	Object Oriented Programming (JAVA) Laboratory	1	
			Total Credit	**20**
Second Semester	**Couse Code**	**Title**	**Credit**	**Remarks**
	CSE 250	Viva-Voce	1	**Theory : 15 Cr.** **Laboratory : 4 Cr.** **Viva-Voce : 1 Cr.**
	STAT 251	Introduction to Probability and Statistics	3	
	CSE 253	Digital Logic Design	3	
	CSE 254	Digital Logic Design Laboratory	1	
	CSE 255	Database Systems	3	
	CSE 256	Database Systems Laboratory	1	
	CSE 257	Algorithms-II	3	
	CSE 258	Algorithms-II Laboratory	1	
	CSE 259	Data and Telecommunication	3	
	CSE 260	Data and Telecommunication Laboratory	1	
			Total Credit	**20**

Third Year				
First Semester	**Couse Code**	**Title**	**Credit**	**Remarks**
	CSE 300	Viva-Voce	1	**Theory : 15 Cr.** **Laboratory : 6 Cr.** **Viva-Voce : 1 Cr.**
	ECO 301	Economics	3	
	CSE 303	Computer Graphics	3	
	CSE 304	Computer Graphics Laboratory	1	
	CSE 305	Computational Geometry	3	
	CSE 307	Computer Architecture and Organization	3	
	CSE 309	Operating Systems	3	
	CSE 310	Operating Systems Laboratory	1	
	CSE 312	Web Design and Programming Laboratory-I (PHP/C#)	2	
	CSE 314	OOAD Laboratory	2	
		Total Credit		**22**
Second Semester	**Couse Code**	**Title**	**Credit**	**Remarks**
	CSE 350	Viva-Voce	1	**Theory : 15 Cr.** **Laboratory : 3 Cr.** **Viva-Voce : 1 Cr.**
	BIS 351	Management and Accounting	3	
	CSE 353	Human Computer Interaction	3	
	CSE 355	Introduction to Bioinformatics	3	
	CSE 357	Microprocessors	3	
	CSE 358	Microprocessors and Assembly Language Laboratory	1	
	CSE 359	Computer Networks	3	
	CSE 360	Computer Networks Laboratory	1	
	CSE 362	Web Design and Programming Laboratory-II (JSP/Python)	1	
		Total Credit		**19**

Forth Year				
First Semester	**Couse Code**	**Title**	**Credit**	**Remarks**
	CSE 400	Viva-Voce	1	**Theory : 12 Cr.** **Laboratory : 4 Cr.** **Research Project : 2 Cr.** **Viva-Voce : 1 Cr.**
	CSE 401	Theory of Computation and Compiler Design	3	
	CSE 403	Software Engineering and Information System Design	3	
	CSE 404	Software Engineering and ISD Laboratory	1	
	CSE 405	Digital Image Processing	3	
	CSE 406	Digital Image Processing Laboratory	1	
	CSE 407	Wireless Networks	3	
	CSE 410	Mobile Application Development Laboratory	2	
	CSE 440	Research Project	2	
		Total Credit		**19**
Second Semester	**Couse Code**	**Title**	**Credit**	**Remarks**
	CSE 450	Viva-Voce	1	**Theory : 12 Cr.** **Laboratory : 5 Cr.** **Research Project : 3 Cr.** **Viva-Voce : 1 Cr.**
	CSE 451	Data Mining and Big Data Analysis	3	
	CSE 452	Data Mining and Big Data Analysis Laboratory	1	
	CSE 453	Artificial Intelligence	3	
	CSE 454	Artificial Intelligence Laboratory	1	
	CSE 455	Software Quality Assurance	3	
	CSE 457	Machine Learning	3	
	CSE 457	Machine Learning Laboratory	1	
	CSE 460	IoT Laboratory	2	
	CSE 480	Research Project	3	
		Total Credit		**21**

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 First Semester

Course code	**:**	CSE 100	**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

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

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

Course code	**:**	MATH 101	**Credit**	**:**	3.0
Title	**:**	Mathematics I (Matrix, Differential Calculus and Coordinate Geometry)	**Prerequisite**	**:**	None
Type	**:**	*Theory*	**Contact hours**	**:**	39

Rationale

The solution of systems of linear equations can be solved by Matrix. Differential calculus concerned with finding tangent lines and rates of change. Algebraic problem can be solved by geometry. So Matrix, Differential calculus and Coordinate geometry is essential for CSE graduates.

Course Objectives

- To learn about matrix, determinant and its application to the solution of systems of linear equations.
- To analyze functions and their graphs. It will be shown how calculus and graphing utilities, working together, can provide most of the important information about their behavior of functions.
- To provide students with a good understanding of the concepts two dimensional geometry.
- To Students will learn about co-ordinate, translation and rotation of axis, conic, conic section for two dimension.
- To learn about two/ three dimensional co-ordinate, conicoid i.e. sphere, cone, cylinder, parpabolid, ellipsoid, hyperboid

Students Learning Outcomes

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

- Use matrix to the solution of systems of linear equations.
- Analyze functions and their graphs and how calculus and graphing utilities, working together, can provide most of the important information about their behavior of functions.
- Understand co-ordinate and its translation and rotation, point, line, mid-point, distance and vector in two-dimensional space.
- Solve two/ three dimensional co-ordinate, conicoid i.e. sphere, cone, cylinder, parpabolid, ellipsoid, hyperboid problems

Course Description

#	Title and Descriptions
1	**Matrices** Definition of matrix; Different types of matrices; Algebra of matrices; Adjoint and inverse of a matrix; Elementary transformations of matrices; Matrix polynomials; Calay-Hamilton theory with uses of rank and nullity; Normal and canonical forms; Solution of linear equations; Eigenvalues and eigenvectors.
2	**Basic differentiation** Graphs and Equations: Functions and Models, Finding Domain and Range, Slope and Linear Functions, Nonlinear Functions and Models, Mathematical Modeling and Curve Fitting, Limits: A Numerical and

	Graphical Approach, Algebraic Limits and Continuity, Average Rates of Change, Differentiation Using Limits of, Difference Quotients, Differentiation Techniques: The Power and Sum–Difference Rules, Differentiation Techniques: The Product and Quotient Rules, The Chain Rule, Higher-Order Derivatives
3	**Application of Differentiation** Using First Derivatives to Find Maximum and Minimum Values and Sketch Graphs, Using Second Derivatives to Find Maximum and Minimum Values and Sketch Graphs, Graph Sketching: Asymptotes and Rational Functions, Using Derivatives to Find Absolute Maximum and Minimum Values, Maximum–Minimum Problems; Business and Economics Applications, Marginals and Differentials, Implicit Differentiation and Related Rates
4	**Exponential and logarithmic Differentiation** Exponential Functions, Logarithmic Functions, Applications: Uninhibited and Limited Growth Models, Applications: Decay, The Derivatives of and a^x and $\log_a{}^x$, Economics Applications
5	**2D Co-ordinate Geometry-1** Change of axes: transformation of coordinates. Simplification of equations of the curves. Pair of straight lines: Homogeneous second degree equations. Conditions for general second degree equations to represent a pair of straight lines. Angle between the lines. Pair of straight lines joining the origin to the points of intersection of the curve and a line.
6	**2D Co-ordinate Geometry-2** Circles and system of circles: Tangents and normals. Pair of tangents. Chord of contact. Orthogonal circles. Radical axis and its properties. Parametric coordinates.
7	**3D Co-ordinate Geometry** Rectangular coordinates. Direction cosines and angle between two lines. The plane and the straight lines. The equation of a sphere. The standard forms of equations of the central conicoids, cones and cylinders.

Recommended Books

1.	Calculus and its applications	Marvin L. Bittinger, David J. Ellenbogen and Scott A. Surgent	10th	Pearson	2012
2.	Textbook of Differential Calculus	Ahsan Akhtar & Sabiha Ahsan	2nd	PHI	2009
3.	Golden Co-ordinate Geometry	N. P. Bali	2nd	LUXMI	2008

Course code	**:**	ENG 103	**Credit**	**:**	3.0
Title	**:**	Communicative English	**Prerequisite**	**:**	None
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

For effective communication, competence in English language skill is necessary. This course gives opportunity to the students to know English grammars and to improve vocabulary, reading, writing, speaking and listening skils.

Course Objectives

- To learn from real life interaction which can help to reinforce the value of their studies.
- To explore the Communicative Approach and how to improve your English Communication skills
- Give students of an international community accurate and meaningful communicating skills which will include expressions for personal identification
- Deals with the practical and communicative aspects of the English Language by reinforcing and manipulating the sounds and grammatical patterns of the language needed in an international situation through dialogues

Students Learning Outcomes

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

- Cope students of an international community accurate and meaningful communicating skills which will include expressions for personal identification
- Deal with the practical and communicative aspects of the English Language by reinforcing and manipulating the sounds and grammatical patterns of the language needed in an international situation through dialogues

Course Description

#	Descriptions
1	**Vocabulary** Clues to the meaning of a word: Position in the clause, prefixes, suffixes, roots, revising and expanding vocabulary, Borrowing, New words from English: compounds and derivatives, meaning and formal relationship
2	**Grammar** Clause: structure, function, variation and expansion, The noun in the clause: number, determiners, The Pronoun in the clause: number, case, agreement and reference. The verb in the clause: form, tense, voice, mood, subject-verb agreement. The modifiers in the clause: adjective, adverb, infinitive, participles, The conjunctions and prepositions to suggest different relationships: time, space, cause, result, purpose, condition, exception, etc., Remedial grammar: Identifying and correcting errors and weaknesses.
3	**Speaking** The art of speaking, body language, how to ask questions, make requests and give instructions; How to respond to queries, invitations and statements; How to introduce and thank, express gratitude, regret or appreciation; How to communicate in particular everyday situations; How to express different concepts: ability, possibility, futurity, necessity, obligation, assumption, regularity, continuity, arrangement, comparison, etc.
4	**Reading** For skimming; For comprehension; For interpretation, Eye movement, chunking, speed reading and SQ3R method
5	**Writing** Spelling, punctuation, indenting, brackets, abbreviation, numbers and fractions, capitalization, underlining, hyphenation, etc, Organization of writing- of sentences in paragraph, and of paragraphs, in essays and letters. Practical Writing**:** personal & official correspondence, job application, CV.

6	**Listening** Lynchpin of Communication, Hearing and Listening, kind of listening, active listening, good listening, barriers of good listening negotiation skills, Introducing audio visual materials and/or movies to develop listening skills.
7	**Creativity and Inter-personal skills** Creativity: Times When you Are Creative, Ways in Which You can be Creative, Developing Your Creativity, Factors that Block Creativity, Mind-Mapping and the Learning Process, Team and Conflict Management, Communication in Teams, Group Discussions (GD), Structuring the GD, Interviews, Techniques of Interviewing, Preparing for an Interview, Kinds of Questions Expected at Interviews, The Interview Process

Recommended Books

1.	Communication Skills for Engineers	Sunita Mishra, C. Muralikrishna	2nd	Pearson	2011
2.	Applied English Grammar & Composition	P.C. Das	4th		1997
3.	How to Speak and Write Correctly	Joseph Devlin	2nd	McGraw-Hill	2013
4.	English Skills with Readings	John Langan , Zoe Albright	9th		2014

Course code	**:**	CSE 105	**Credit**	**:**	3.0
Title	**:**	Structured Programming	**Prerequisite**	**:**	None
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

This course is designed to introduce students in the algorithmic way of thinking and problem solving by computers. Students learn the fundamental principles of structured programming. Typical characteristics and mechanisms of a structured programming language are introduced and students are introduced to the design and development of structured programs in this language. C programming language is used as the course basis. Lectures are completed by lab practice where theoretical knowledge is applied in an appropriate software environment.

Course Objectives

- To make students familiar with basic programming principles, good programming style, structured approach to program design, development, testing and documentation

Students Learning Outcomes

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

- Formulate problems step by step and design computer programs in a structured way
- Identify programming data structures and describe programming methodologies

- Apply fundamental programming concepts using high-level programming language to solve problems

Course Description	
#	**Title and Descriptions**
1	**Structured Programming Language fundamentals** C overviews, History and Features, Basic Structure of C Program and Hello World Program, C Program Development Environment
2	**Variables, Constants, Data Types, Operators & Expression** Declaring variables and assigning values, input from keyboard, add comments, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operators, Bitwise Operators, Special Operators, Arithmetic Expressions, Evaluation of Expressions, Type Conversions in Expressions, Operator Precedence and Associativity.
3	**Program control statements** Decision Making Statements: if-else statement, switch statement; Looping Statements: for loop, while loop, nested if, do while loop, nested loop; Jump Statements: continue, break
4	**Functions** Function prototype, recursion, parameters, arguments, scope rules and storage classes.
5	**Arrays and Pointer** One and Multi-dimensional arrays, Character Arrays and Strings, Basic of Pointer, pointer expression, pointer arrays.
6	**User defined data types and Input/ Output** Structures, Unions, Enumerations, Standard input and output, Formatted input and output, File access; Variable length argument list; Command line parameters; Error Handling; Graphics; Linking; Library functions.
7	**Memory manipulation and Preprocessor** Dynamic Memory Allocation and Linked List, Macro substitution, File inclusion, Compiler Control Directives.

Recommended Books					
1.	Teach yourself C	Herbert Schildt	3rd	McGraw-Hill	1997
2.	Programming in ANSI C	E. Balagurusamy	7th	Tata Mcgraw-Hill	2016
3.	C: The Complete Reference	Herbert Schildt	4th	McGraw-Hill	2000
4.	C-How to Program	Paul Deitel and Harvey Deitel	7th	Prentice Hall	2017

Course code	**:**	CSE- 106	**Credit**	**:**	1
Title	**:**	Structured Programming Laboratory	**Prerequisite**	**:**	None
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale
This lab course is designed for the students to achieve a hands on experience on basic programming. C programming language is used as the course basis. Theoretical lectures are completed by lab practice where theoretical knowledge is applied in an appropriate software environment.

Lab Objectives
• To introduce students to give practical experience on basic programming principles, good programming style, design and solve the problems in a structured approach.

Lab Outcome
Upon successful completion of this course, the students should be able to: • Solve problems step by step and design computer programs • Apply fundamental programming concepts using high-level programming language

Lab Course Description

Exp. #	**Title and Descriptions**
1	Introduction: - Code::Block or Dev-C++ or Visual C++ - How to write, save, compile and run a C program.
2	Variables, Constants, Data Types, Operators & Expression (PART-I)
3	Variables, Constants, Data Types, Operators & Expression (PART-II)
4	Managing Input and Output Operations and Conditional Statements
5	Looping Statements
6	Arrays and String
7	Search and Sorting
8	User-defined Functions
9	Recursion
10	Pointer
11	Structure and Union and File processing
12	Dynamic Memory Allocation and Linked list
13	Graphics Programming

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
Core i5, 1.8 GHz, 4 GB RAM, 500 MB disk space	**Popular C compilers/IDEs include:** Code::Block/ Microsoft Visual Studio Community

Course code	**:**	CSE- 107	**Credit**	**:**	3.0
Title	**:**	Electrical Circuits	**Prerequisite**	**:**	None
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

The course gives an introduction to electrical circuit theory. It provides a fundament for understanding and designing simple circuits and systems built with analog electrical circuit elements.

Course Objectives

- To use basic circuit theory to solve problems in electrical engineering and analyze/design simple circuits
- To use laboratory equipment such as voltmeter, ammeter, oscilloscope and signal generator
- To describe the most important components and functioning of the power distribution network
- To have knowledge about the energy production using, e.g., solar cells, and basic DC motors/generators

Students Learning Outcomes

Upon successful completion of this course, students will be able to-

- understand the basic theory and mathematical relationships in circuit analysis
- understand basic terms and results from the theory about circuits with resistances, capacitances and inductance components, as well as basic semiconductor components
- know the principles for ideal circuits and power grid components

Course Description

#	Descriptions
1	**Network sources and Analysis** Voltage sources; Voltage sources in parallel; Open and short circuits; Current sources in series and parallel; Mesh analysis; Nodal analysis; Star-delta and delta-star conversion.
2	**Basic Passive Elements** Resistors, inductors and capacitors in series and parallel; Transient response in capacitive networks; Charging and discharging phases; R-L transients; Storage cycle, Decay phase.
3	**Network Theorems 01** Superposition theorem; Thevenin's theorem; Norton's theorem;
4	**Network Theorems 02** Maximum power transfer theorem; Millman's theorem; Substitution theorem; Reciprocity theorem.
5	**AC: Part I** Fundamentals of AC and the Basic elements and Phasors: Generation of alternating voltage & currents; Sine wave; General format of sinusoidal voltage and currents; Phase & algebraic representations of sinusoids; Average & RMS (effective) values;
6	**AC: Part II**

	Response of basic R,L,C elements to a sinusoidal voltage & currents; frequency response of basic elements; Resonance; Average power & power factor; Complex numbers: Rectangular & polar form: Active & reactive power; Series & parallel resonance circuit; Qualify factor, Selectivity.
7	**Transformer** Construction and features of transformer; Transformer on no- load and on load; emf-equation; Phasor diagram; Equivalent circuits; Losses and efficiency.

Recommended Books

1.	Introductory Circuit Analysis	R. L. Boylestad	10th	Prentice Hall	2010
2.	Alternating Current Circuits	R. M. Kerchner, G. F. Corcoran	7th	Wiley	2010
3.	Electrical Circuits: An Introduction	K. C. A. Smith, R. E. Alley	2nd	University of Cambridge	1992
4.	Electric Machines	D. P. Kothari, I. J. Nagrath	4th	Tata McGraw-Hill	2004

Course code	**:**	CSE 108	**Credit**	**:**	1
Title	**:**	Electrical Circuits laboratory	**Prerequisite**	**:**	None
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

Through examples and laboratory exercises the students should achieve a practical knowledge of analog electronic circuit elements, exercise in use of basic laboratory equipment and an introduction to writing laboratory reports.

Lab Objectives

- to use laboratory equipment such as voltmeter, ammeter, oscilloscope and signal generator
- to describe the most important components and functioning of the power distribution network by experiment
- To have knowledge about the energy production using, e.g., solar cells, and basic DC motors/generators

Lab Outcome

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

Lab Course Description

Exp. #	Title	Contact Hours
1	Familiar with Electrical Lab components and Equipment	3
2	Measurement of DC Voltage and Current	3
3	Computer Analysis of Electrical Circuit	3

4	Measurement of Instrument Characteristics	3
5	Δ/Y Conversions and the Wheatstone Bridge	3
6	Network Theorems-I	3
7	Network Theorems-II	3
8	Investigation of Digital Oscilloscope and Measurement	3
9	Observation of RC and RL circuit	3

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
Ammeter, Voltmeter, Oscilloscope, Power Supply, Pcs, Registers, Capacitors, Inductors, Transformers	OrCad software

---

Course code	**:**	PHY- 109	**Credit**	**:**	3.0
Title	**:**	Physics	**Prerequisite**	**:**	None
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

The course gives an introduction to electricity and magnetism and optics. It provides a fundament for understanding and charge, electric filed, force on a current, dispersion and electrical and mechanical measurement.

Course Objectives

- To use basic circuit theory of charge, electric field and laws of electrical simple circuits
- To use laboratory equipment such as voltmeter, ammeter, oscilloscope and signal generator
- To describe the basic circuit theory of Magnetic field, force on current and magnetic properties of matter.

Students Learning Outcomes

Upon successful completion of this course, students will be able to-

- understand basic circuit theory of charge, electric field and laws of electrical simple circuits
- understand the basic circuit theory of Magnetic field, force on current and magnetic properties of matter
- know the principles of optics.

Course Description

#	**Descriptions**
1	**Charge, Electric field & Gauss's Law-I** Simple phenomena in electrostatics; Electrostatic induction and charge density; Coulomb's law; Electric field & field strength; Point charge in an electric field;

2	**Charge, Electric field & Gauss's Law-II** Dipole in an electric field; Electric flux; Gauss's law and some applications; Electric potential; Potential due to a point charge; Equipotential surfaces; Potential energy; Potential gradient; Capacitance and its calculation; Parallel plate capacitor with dielectric; Dielectric & Gauss's law; Electric vectors; Energy stored in an electric field
3	**Electric current, Simple circuits and Electrical Measurements-I** Current and Ohm's law; E.M.F. and potentiaFdifference; Kirehhoffs laws; Whetstone bridge; Single loop & multi loop circuits; Simple RC and RL circuits, Kirchhoff's laws.
4	**Electric current, Simple circuits and Electrical Measurements-II** The potentiometer; Moving coil Galvanometer; Ammeter; Voltmeter; Multimeter; Wattmeter & Energy meter; Measurements of Voltage, Current, Resistance, Inductance, Capacitance, Power and Energy.
5	**Magnetic Field & Force on Current** Coulomb's law; Magnetic field and field strength; Magnetic force on current; Ampere's law; Directions of current and field; Maxwell's cork screw rule; Fleming's left hand rule; Magnetic field near long wire; Magnetic field for solenoid; Biot-savart law. Faraday's law of electromagnetic induction; Fleming's right hand rule; Lenz's law.
6	**Magnetic properties of matter** Poles and dipoles; Coulomb's law for magnets & Gauss's theorem of magnetism; Dia-magnetism, Para-magnetism and Ferro-magnetism; Magnetomotive force and field intensity; Concept of self and mutual inductance; Coefficient of magnetic coupling; Rise of current and decay of current in Inductive circuit; Energy in magnetic field; Inductance in series and parallel; Hysteresis and eddy current losses.
7	**Optics** Refraction and total internal reflection; Group velocity and Phase velocity of light; Dispersion; Interference; Holography; Fresnel and Fraunhofer diffraction; Polarization of light wave.

Recommended Books

1.	Physics Part-II	DavidHalliday, Robert Resnick	2nd	Wiley Eastern Limited	1992
2.	Electricity and Magnetism	Edward M. Purcell, David J. Morin	3rd	Cambridge University Press	2013
3.	Introduction to Modern Optics	Grant R. Fowles	2nd	Dover Publications	1989
4.	Electricity, Magnetism, and Light	Wayne M. Saslow	1st	Academic Press	2004

Course code	**:**	URP 112	**Credit**	**:**	1
Title	**:**	Computer Aided Engineering Drawing Laboratory	**Prerequisite**	**:**	None
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Lab Objectives

- To enable students to produce 2D and 3D engineering drawings using CAD tools.

Lab Outcome

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

- Produce 2D and 3D engineering drawings using CAD tools

Lab Course Description

#	Title
1	Getting familiar with the Auto CAD Environment. Toolbars, working area, sub menus, working modes. Starting with some basic commands.
2	Study addressing schemes with different commands
3	Studying basic objects and their commands e.g. circle, donut rectangle, arc, ellipse, polygon
4	Studying commands that duplicate objects e.g. array, offset and modify commands e.g. trim, break, chamfer, fillet.
5	Studying Mirror, hatch, ltype, adding toolbars and object snap, zoom, text
6	Making Isometric objects with isometric settings
7	Applying dimensions (Aligned, Radius, Diameter, Angular, Leaders). Increasing / Decreasing working area, changing measuring scales
8	Changing properties of dimensions through style. Modifying properties of objects. Changing dimensions using stretch and extend
9	Changing views for 3d drawings, studying Solids and 3d objects box, sphere, cylinder, cone, wedge, torus, extrude, revolve

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
PCs	AutoCAD software

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

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

Second Year First Semester

Course code	**:**	CSE 200	**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 second year first semester.

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

Course code	:	MATH 201	**Credit**	:	3.0
Title	:	Mathematics III (Vector, Complex Variable, Fourier Analysis and Laplace Transformation)	**Prerequisite**	:	None
Type	:	*Theory*	**Contact hours**	:	39

Rationale

Vector spaces are widely used in Data Science. Complex numbers are used to encode geometric information through algebra. The Laplace transform has a number of properties that make it useful for analyzing linear dynamical systems. Most of the works of image matching, face recognition, 3-D map creation, video-processing, sound processing are done using Fourier Transform, Laplace Transform. Fourier series help to better analyze a signal in another domain rather in the original domain.

Course Objectives

- Introduce and develop the methods of vector analysis.
- Introduce vector spaces and linear transformations, and their applications to a variety of problems.
- Introduce complex number system in details.
- Provide basic concept of Fourier series, Fourier integral, Fourier transforms, Laplace Transforms and Inverse Laplace Transforms of different functions.

Students Learning Outcomes

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

- Explain the basic concepts of vector spaces and subspaces.
- Demonstrate an understanding of the fundamental concepts of complex analysis.
- Apply theorems and rules in problem analysis.
- Calculate Fourier and Laplace Transforms and Inverse Laplace Transforms.
- Solve boundary problems for using Fourier Transform.
- Classify Laplace Transforms, Definition, existence and basic properties, differential function, periodic function etc.
- Describe different types Inverse Laplace Transform with the help of partial fraction, evaluation of integrals.

Course Description

#	Title and Descriptions
1	**Vector Spaces** Definition and properties, subspaces, basis and dimension, change of basis; Linear Transformation (LT): definition and properties, linear operator matrix, geometry of LT, standard plane LT.
2	**Vector Calculus** Differentiation and integration of vectors together with elementary applications; Definition of line, surface and volume integrals; Gradient, divergence and curl of point functions, various formulae, Gauss's theorem,

	Stoke's theorem, Green's theorem.
3	**Complex Variable** Complex number system; General functions of a complex variable; Limits and continuity of a function of complex variable and related theorems; Complex differentiation and the Cauchy's Riemann Equations; Mapping by elementary functions.
4	**Integral Theorem** Line integral of a complex function; Cauchy's Integral Theorem; Cauchy's Integral Formula; Liouville's Theorem; Taylor's Theorem and Laurent's Theorem. Singular points; Residue; Cauchy's Residue Theorem. Evaluation of residues; Contour integration; Conformal mapping.
5	**Fourier Analysis** Real and complex form of Fourier series; Finite transform; Fourier Integral; Fourier transforms and their uses in solving boundary value problems of wave equations.
6	**Laplace Transforms** Definition; Laplace transforms of some elementary functions; Sufficient conditions for existence of Laplace transforms; Inverse Laplace transforms; Laplace transforms of derivatives.
7	The unit step function; Periodic function; Some special theorems on Laplace transforms; Partial fraction; Solutions of differential equations by Laplace transforms; Evaluation of improper integrals.

Recommended Books					
1.	Vector and Tensor Analysis	George E. Hay	1st	Dover Publications	2012
2.	Schaum's Outline Complex Variables	Murray Spiegel, Seymour Lipschutz, John Schiller, Dennis Spellman	2nd	McGraw-Hill	2009
3.	Fourier Analysis and Its Applications	Gerald B. Folland	2nd	American Mathematical Society	2009
4.	An Introduction to Laplace Transforms and Fourier Series	Phil Dyke	2nd	Springer	2014

Course code	:	CSE 203	**Credit**	:	3.0
Title	:	Computer Ethics and Cyber Law	**Prerequisite**	:	N/A
Type	:	Theory	**Contact hours**	:	39

Rationale

In the recent years, many concerns and issues were raised on the integrity and security of information, legal status of online transactions, privacy and confidentiality of information, intellectual property rights and security of government data placed on the Internet. Therefore, ethical values in computing are essential for understanding and maintaining the relationship between computing professionals and researchers and the users of their applications

and programs. The concepts are needed to understand risks and how to deal with them. Since computer ethics and cyber law are constantly changing as technology changes, there is a need for understanding the concept of risks and how to deal with them.

Course Objectives

- Intended to give students a chance to reflect on the humanitarian, social, and professional impact of computer technology by focusing on ethical issues faced by and brought about by computing professionals, including those related to networking and the internet, intellectual property, privacy, security, reliability, and liability.
- Introduce the students to professional ethics, codes of conduct, and moral responsibility
- To facilitate understand & critical understanding about Cyber crimes, Ethical Hacking, cyber security, forensics and cyber laws
- To provide an understanding of principal concepts, major issues, technologies and basic approaches in cyber security.
- Focus on issues raised by the possible emergence in the future of highly intelligent machines.

Students Learning Outcomes

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

- Understand, identify, and apply different ethical philosophies, frameworks, and methodologies.
- Identify and interpret the codes of professional conduct relating to the disciplines of computer science and software engineering.
- Analyze the local and global impact of computing on individuals, organizations, and society.
- Understand and apply the concepts and principles of moral thinking to problems relating to computing and digital technologies.
- Improve their skills writing argumentative essays and pieces, and in critical thinking, analysis, and presentation.
- Become familiar with a number of noteworthy essays written by influential researchers in the field of cyberethics.

#	Title and Descriptions
1	**Introduction** What is Cyber Law, Cyber Space, Various Cyber Crimes, Legal fabric of Bangladesh regarding cyber-crimes, What is Computer Ethics, Ten Commandments of computer ethics, Code of ethics for information, technology professionals.
2	**Ethics for IT Professionals and IT Users** Definition of IT Professionals, Relationships Between IT Professionals and Employers, Clients, Suppliers, Other Professionals, IT Users and Society, Certification, Government Licensing, Common Ethical Issues for IT Users, Supporting the Ethical Practices of IT Users.
3	**Computer and Internet Crime**

	Why Computer Incidents Are So Prevalent, Types of Exploits, Types of Perpetrators, Types of Perpetrators, Establishing a Security Policy, Prevention, Detection, Response, and Computer Forensics.
4	**Privacy** The Right of Privacy, Data Encryption, Public Key Encryption, Private Key Encryption, Identity Theft, Phishing, Spyware, Spamming, Advanced Surveillance system.
5	**Intellectual Property and Software Development** Definition of Intellectual Property, Copyrights, Patents, Trade Secret Laws, Plagiarism, Reverse Engineering, Open Source Code, Competitive Intelligence, Strategies to Engineer Quality Software, Software Development Process, Development of Safety-Critical Systems, Quality Management Standards.
6	**National Cyber security Strategy** Legal Measures, Technical and Procedural, Organizational Structures.
7	**Association for Computing Machinery (ACM) Code of Ethics and Professional Conduct** General Moral Imperatives, Organizational Leadership Imperatives, Compliance with the Code

Recommended Books					
1.	Ethics and Technology	Tavani, H.T.	4th	Wiley	2011
2.	Ethics for the Information Age	M. J. Quinn	7th		
3.	Computer Ethics and Professional Responsibility	Terrell Ward Bynum, Simon Rogerson	Latest edition	Wiley	2003

Course code	**:**	CSE-205	**Credit**	**:**	3.0
Title	**:**	Numerical Methods	**Prerequisite**	**:**	N/A
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Numerical error and stability analysis is the key to develop accurate and robust algorithms. Numerical methods, based upon sound computational mathematics, are the basic algorithms underpinning computer predictions in modern systems science. The subject is initiated with fundamental principles of digital computing and the implications for algorithm accuracy and stability. The solution of systems of linear equations, the error and stability issues associated with solving linear systems will be covered extensively. The numerical treatment of eigenvalue problems is briefly discussed. Several lectures are devoted to solving non-linear equations, including root finding. The concept of interpolation and its role as foundation for numerical differentiation and integration is introduced. Numerical differentiation and integration is covered in depth, with particular emphasis on the error and convergence analysis. The final part of the course introduces the fundamentals of finite-difference solutions to ordinary differential equations, again with emphasis on error and convergence analysis.

Course Objectives

- To introduce students with different numerical methods for solving real life mathematical, physical and engineering problems.
- To give basic understanding of how to solve system of linear equations.
- To provide basic knowledge of curve fitting.
- To give a basic knowledge of numerical integration, and numerical differentiation.
- To provide introductory concepts of differential equations.
- To prepare students for facing future challenges.

Students Learning Outcomes

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

- Apply numerical approaches to solve system of linear equations.
- Apply numerical techniques to solve integration and differentiation.
- Explain the concept of curve fitting and partial differential equations.

#	Title and Descriptions
1	**Numerical Computing Process and Numerical Errors** Definition, Necessity, Process of Numerical Computing, Characteristics of Numerical Computing. Significant Digits, Errors in Arithmetic, Different types of Errors in Numerical Computation.
2	**Roots of Nonlinear Equations** Definition, Necessity, Methods of Solution, Iterative Methods: Bisection, False position, Newton-Raphson, Secant, and Fixed Point method.
3	**Direct and Iterative Solution of Linear Equations** Need and Scope, Existence of Solution, Basic Gaussian Elimination Method, Gauss-Jordan Method, and Matrix Inversion Method. Jacobi Iteration Method and Gauss-Seidel Iteration Method.
4	**Curve Fitting** Interpolation Definition, Need and Scope, Linear Interpolation, Lagrange Interpolation Polynomial and Newton Interpolation Polynomial, Divided Difference Table. Regression Definition, Need and Scope, Fitting Linear Equations- Least Square Regression, Fitting Transcendental Equations.
5	**Numerical Differentiation** Need and Scope, Differentiating continuous functions, Higher-Order Derivatives, Differentiating tabulated functions.
6	**Numerical Integration** Need and Scope, Trapezoidal rule, Simpson's 1/3 rule and Simpson's 3/8 rule.
7	**Numerical Solution of Ordinary and Partial Differential Equation** Need and Scope, Taylor series Method, Picard Method, Euler's method, Heun's Method, Runge-Kutta Method, Determination of characteristics equation of a matrix using Fadeev-Leverrier method, Eigen value and Eigen vector and matrix inversion.

Recommended Books					
1.	Numerical Methods	E Balagurusamy	8th	Tata McGraw-Hill	
2.	Introductory Methods of Numerical Analysis	S. S. Sastry	5th	PHI Learning Private Ltd.	
3.	Applied Numerical Method for Engineers	Robert J. Schilling and Sandra L Harries	5th	Thomson	
4.	Numerical Methods For Engineers and Scientists: An Introduction with Applications Using MATLAB	Amos Gilat, Vish Subramaniam	2nd	Wiley	
5.	Numerical Methods for Engineers	Steven C. Chapra and Raymond P. Canale	6th	McGraw-Hill	

Course code	**:**	CSE-206	**Credit**	**:**	1
Title	**:**	Numerical Methods Laboratory	**Prerequisite**	**:**	
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Lab Objectives

- To provide a basic knowledge of programming in C and MATLAB.
- To give a practical experience of how to implement different numerical methods.
- To prepare students for future laboratory-oriented courses.

Lab Outcome

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

- Write algorithms and flowcharts of different numerical methods.
- Write programs in C and MATLAB to implement numerical methods for solving real life problems.

Exp. #	Title
1	Implementation of Bisection Method.
2	Implementation of False Position Method.
3	Implementation of Newton-Raphson Method.
4	Implementation of Jacobi Iteration Method.
5	Implementation of Gauss-Seidel Iteration Method.
6	Implementation of Lagrange Interpolation Polynomial Method.
7	Implementation of Newton Interpolation Polynomial Method.
8	Implementation of Least Square Regression Method.
9	Implementation of Trapezoidal rule.

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
PCs	1. C/C++ /MATLAB and MS Excel.

Course code	:	CSE-207	**Credit**	:	3.0
Title	:	Electronic Devices and Circuits-II	**Prerequisite**	:	CSE-157
Type	:	Theory	**Contact hours**	:	39

Rationale

This course is designed to understand the two terminal devices, theory and operation of the operational amplifier, applications and Power amplifiers, Multivibrators, Oscilators and feedback amplifiers

Course Objectives

- To understand the two terminal devices, theory and operation of the operational amplifier, applications and Power amplifiers, Multivibrators, Oscilators and feedback amplifiers

Students Learning Outcomes

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

- Operating operational amplifier, applications and Power amplifiers, Multivibrators, Oscilators and feedback amplifiers.

#	Descriptions
1	**Two terminal Devices** Schottky Barrier (Hot-Carrier) Diodes, Varactor (Varicap) Diodes, Solar Cells, Photodiodes, Photoconductive Cells, IR Emitters, Liquid-Crystal Displays, Thermistors, Tunnel Diodes
2	**Operational Amplifier** Differential Amplifier Circuit, BiFET, BiMOS, and CMOS Differential Amplifier Circuits, Op-Amp Basics, Practical Op-Amp Circuits, Op-Amp Specifications—DC Offset Parameters, Op-Amp Specifications—Frequency Parameters, Unit Specifications, Differential and Common-Mode Operation
3	**Op-Amp Applications** Constant-Gain Multiplier, Voltage Summing, Voltage Buffer, Controlled Sources, Instrumentation Circuits, Active Filters, Differential amplifier; Stable ac coupled amplifier; Analogue integrator & differentiator; Logarithmic and anti-logarithmic amplifiers; Multipliers; Dividers; Squarer & square rooters; Electronic analogue computation.
4	**Power Amplifiers** Class A, Class B & Class C amplifiers, R-C coupled, Direct coupled and Transformer coupled amplifier, push-pull, complementary symmetry amplifier, simple tuned, inductively tuned and double tuned amplifier, power amplifier, wide band amplifier.
5	**Multivibrators** Concepts, Monostable, Astable and Bistable multivibrators using transistor & OP-AMPs; Schimitt trigger circuits; The 555 timer.

6	**Oscillators** Concept; Circuit requirement for oscillation; Nyquist criterion; Sinusoidal oscillators; Barkhausen criterion; Phase shift oscillator; Resonant circuit oscillator; A general form of oscillator circuit; Hartley & Colpitts oscillator; Wein bridge oscillator; Crystal oscillator.
7	**Feedback Amplifiers** Feedback Concepts, Feedback Connection Types, Practical Feedback Circuits, Feedback Amplifier—Phase and Frequency Considerations

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 208	**Credit**	**:**	1
Title	**:**	Electronic Devices and Circuits-II Laboratory	**Prerequisite**	**:**	CSE-157
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

This course is designed to understand the two terminal devices, theory and operation of the operational amplifier, applications and Power amplifiers, Multivibrators, Oscilators and feedback amplifiers

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 two terminal devices (Varactor, Solar cell and Photodiodes).
2	Study of the characteristics of two terminal devices(IR Emitters, Liquid-Crystal Displays, Thermistors).
3	Study the behavior of operational amplifier.
4	Study of the applications of operational amplifier.

5	Study of the characteristics of different type power amplifier
6	Study of the characteristics of monostable multivibrator
7	Study of the characteristics of bistable multivibrator
8	Study of the characteristics of oscillators
9	Study of the characteristics of Feedback amplifier.

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-209	**Credit**	**:**	3.0
Title	**:**	Algorithms I	**Prerequisite**	**:**	C/C++/Java
Type	**:**	Theory	**Contact hours**	**:**	41

Rationale

Algorithms are the soul of computing. The construction and analysis of algorithms is a basic and very important part of modern computer science. In mathematics and computer science, an algorithm is a step-by-step procedure which is used for calculation, data processing, and automated reasoning. All computer programs can be described as algorithms that operate on a structured set of data, or as a concatenation of such algorithms. To construct a large program with a reasonable time and space consumption it is essential to have efficient solutions to the problem parts. Algorithms help to acquire necessary skills to recognise problem scenarios and identify the right algorithms that can be used, to modify an existing algorithm or develop a new one for new problems.

Course Objectives

- Provides an introduction to mathematical modeling of computational problems.
- Covers the common algorithms, algorithmic paradigms, and data structures used to solve these problems.
- Emphasizes the relationship between algorithms and programming, and introduces basic performance measures and analysis techniques for these problems.

Students Learning Outcomes

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

- Solve problems using appropriate algorithm.
- Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it.
- Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
- Describe the greedy paradigm and explain when an algorithmic design situation calls for it.
- Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it.

- Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.

#	Title and Descriptions
1	**Introduction to Algorithm** What Kind of Problems are solved by Algorithms?, What is Data Structure?, Algorithms as a Technology, Analyzing Algorithms, Searching, Linear Search, Binary Search, Sorting, Insertion Sort, Selection Sort, Bubble Sort, Algorithmic paradigms
2	**Divide and Conquer** Property, Recurrence, Power of an element, Merge Sort, Counting Inversion in Merge Sort, Quick Sort, Randomized Quick Sort
3	**Sorting** Heap Sort, Linear Sorting Algorithms, Counting Sort, Radix Sort
4	**Greedy Algorithms** Property, Problems, Task Scheduling, Fractional Knapsack, Greedy Coin Change, Huffman Coding
5	**Data Structures** Disjoint Set Union, Binary Search Tree
6	**Dynamic Programming** Property, Problems : Fibonacci Number, Rod Cutting, Knapsack, Coin Change, Hill Climbing, LCS, Bit-Masking
7	**Graph Algorithms** Graph Representation, BFS, DFS, Topological Sort, Shortest Path, Single Source Shortest Path: Dijkstra Algorithm, Bellman Ford Algorithm, All Pair Shortest Path: Floyd Warshall, Johnsons Algorithm, Minimum Spanning Tree: Prims Algorithm, Kruskal Algorithm.
8	**Number Theory** Elementary number-theoretic notions, GCD, Modular Arithmatic, Solving Modular linear equations, Prime factorizations, Chinese Remainder Theorem.

Recommended Books

1.	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	3rd	The MIT Press	
2.	Algorithms Unlocked	Thomas H. Cormen	1st	The MIT Press	
3.	Introduction to the Design and Analysis of Algorithms	Anany Levitin	3rd	Pearson	

Course code	**:**	CSE-210	**Credit**	**:**	1
Title	**:**	Algorithm-I Laboratory	**Prerequisite**	**:**	C / C++ / Java
Type	**:**	Laboratory Work	**Contact hours**	**:**	[52]

Lab Objectives

- Provides an introduction to mathematical modeling of computational problems.
- Covers the common algorithms, algorithmic paradigms, and data structures used to solve these problems.
- Emphasizes the relationship between algorithms and programming, and introduces basic performance measures and analysis techniques for these problems.

Lab Outcome

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

- Solving problems using appropriate algorithm.
- Implement the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
- Implement the greedy paradigm and explain when an algorithmic design situation calls for it.
- Implement the dynamic-programming paradigm and explain when an algorithmic design situation calls for it.
- Implement major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.

Exp. #	**Title**
1	Searching a key in an array a. Linear Search and Insertion Sort b. Selection Sort, Bubble Sort, Binary Search
2	Sorting an array using: Insertion Sort, Selection Sort, Bubble Sort
3	Using divide and conquer approach for sorting: Merge Sort and Quick Sort
4	Using data structure for sorting: Heap Sort, Linear time sorting: Counting Sort, Radix Sort
5	Task Scheduling using Greedy Approach
6	Fractional Knapsack and Greedy Coin Change
7	Huffman Coding and Disjoint Set Union(Union Find)
8	Dynamic Programming
9	Graph Representation and Breadth First Search(BFS)
10	Depth First Search(DFS) and Topological Sort Single Source Shortest Path, Dijkstra Algorithm, Bellman Ford Algorithm

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
Intel® Pentium® 4 Processor	Code Blocks / InteliJ / Net beans / Eclipse

Course code	:	CSE- 212	**Credit**	**:**	1
Title	:	Object Oriented Programming (JAVA) Laboratory	**Prerequisite**	**:**	OOP Concept
Type	:	Laboratory Work	**Contact hours**	**:**	26

Rationale

Java is a programming language, designed to be concurrent, class-based and object-oriented, and specifically designed to have as few implementation dependencies as possible. Its portability, safety, and simplicity features made one of the most popular programming languages. The web acts as convenient transport mechanism for Java programs and the web's ubiquity has popularized Java as an Internet development tool. The underlying principle that has enabled Java's success is the ability of the developers and coders consistently upgrading the model to be competitive to modern technological standards.

Lab Objectives

- To introduce students about basic object oriented programming principles: abstract data types, encapsulation, inheritance and polymorphism in Java.
- To make students familiar with fundamental features of Java: object, classes and interfaces, exceptions and libraries of object collections.
- Equip students with the required object -oriented programming skills required to build highly reusable, robust and maintainable software systems.

Lab Outcome

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

- Solve real world problems using object oriented programming principles
- Create, compile and debug computer programs in JAVA.
- Design and implement classes to produce reliable, robust, and reusable code.
- Implement object-oriented designs using encapsulation, inheritance, polymorphism, and exception handling
- Develop software (application program) in JAVA.

Exp. #	Title
1	**Introduction** Basic structure and Syntax of Java Program, E. Lists and Maps F. Sorting, searching and string
2	**Object-Based Programming** A. Classes and objects, instance variables, and instance methods B. Member access modifiers: public, private, protected, package, Constructors, overloaded constructors D. Set (mutator), Get (access), and predicate methods E. Final instance variables
3	**Object-Based Programming** J. Creating packages (Packages, The import Statement, Static Imports, CLASSPATH and Import, Defining Packages, Package Scope) and Allocation of Project

4	**Object-Oriented Programming** A. Inheritance B. Super class, subclass
5	**Object-Oriented Programming** C. Polymorphism D. Dynamic method binding
6	**Object-Oriented Programming** E. Abstract class, Concrete class F. Inner class definition G. Type-wrapper class for primitive data types H. Interfaces
7	**Exception Handling** A. Exceptions Overview B. Catching Exceptions C. The finally Block D. Exception Methods E. Declaring Exceptions F. Defining and Throwing Exceptions G. Errors and Runtime Exceptions
8	**Input/Output Streams** A. Overview of Streams B. Bytes vs. Characters C. Converting Byte Streams to Character Streams D. File Object E. Binary Input and Output F. PrintWriter Class G. Reading and Writing Objects H. Closing Streams
9	**Java – Thread** A. Multitasking B. Main Thread C. How to create Thread D. Implementing Runnable E. Extending Thread F. Multiple Threads G. Synchronization H. Inter Thread Communication
10	**Graphical User Interface - Java Swing** A. Event-Driven Programming and Event Handling Model B. Window Components C. Mouse and keyboard event handling D. Adapter classes E. Layout managers and Java Database, javado

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
Here are the minimum requirement: Core i5, 1.8 GHz, 4 gig RAM, 500 meg disk space	Here are the minimum requirement: • Java SDK or JRE 1.6 or higher • Supported Database and library that supports the database connection with Java Popular IDEs include: i. Netbeans/ Eclipse

Recommended Books					
1.	Java: The Complete Reference	Herbert Schildt	9th	McGraw-Hill Education	2014
2.	An Introduction to Object-oriented Programming with Java	C. Thomas Wu	5th	McGraw-Hill	2006
3.	Java How to Program	Paul Deitel, Harvey Deitel	9th	Prentice Hall	2011
4.	Head First Java	Kathy Sierra, Bert Bates	2nd	O'Reilly Media	

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

Second Year Second Semester

Course code	**:**	CSE 250	**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 second year second semester.

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

Course code	**:**	STAT 251	**Credit**	**:**	3.0
Title	**:**	Introduction to Probability and Statistics	**Prerequisite**	**:**	None
Type	**:**	*Theory*	**Contact hours**	**:**	39

Rationale

The theory and methods of Statistics play an important role in all walks of life, society, medicine and industry. They enable important understanding to be gained and informed decisions to be made, about a population by examining only a small random sample of the members of that population. The statistical inferences about a population are subject to uncertainty. Probability theory is the branch of mathematics that deals with modelling uncertainty, and so to understand statistics, we must understand uncertainty, and hence understand probability. It is important because of its direct application in areas such as genetics, finance and telecommunications. It also forms the fundamental basis for many other areas in the mathematical sciences including statistics, modern optimisation methods and risk modelling.

Course Objectives

- Basic idea about probability and probability distribution.
- Application of probability and probability distribution in real life example.
- Find the different characteristics of probability distribution.
- Understand the basic concept of sampling and gather knowledge about how to apply different sampling techniques in any sample.
- Introduce the basic data summary techniques, their presentations and interpretation, concept of randomness and how to make inference under these conditions.

Students Learning Outcomes

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

- Identify the role of probability and probability distribution and **c**alculate and interpret probability of any given event.
- Understand underlying concept of r**andom variable and their usage and** laws of probability and the use of Bayes theorem and formulate the concept of a statistical distribution.
- Be able to calculate the distribution of a function of a random variable and to use probability distribution in different practical situation and as well as find different properties of the distribution.
- Use and extend knowledge of statistical inference techniques and their applications in real -life situations.

Course Description

#	Title and Descriptions

1	**Introduction to Statistics and Data Description** Field of Statistics, Data, Graphical Presentation of Data, Numerical Description of Data: Measures of Central Tendency, Measures of Variation, Correlation and Regression Analysis.
2	**Introduction to Probability** Review of set, Experiments and Sample Spaces, Events, Probability Definition and Assignment, Tree Diagram, Multiplication Principe, Conditional probability, total probability, and Bayes' Theorem and its Application to Engineering problems. One-Dimensional Random Variable and its Function. Distribution function, Discrete Random Variable, Continuous Random Variables, Chebyshev's Inequality.
3	**Joint Probability Distributions** Joint Distribution for Two-Dimensional Random Variables, Marginal Distributions, Conditional Distributions, Conditional Expectation, Regression of the Mean, Independence of Random Variables, Covariance and Correlation, the Distribution Function of Two-Dimensional Random Variables, Functions of two Random Variables.
4	**Discrete and Continuous Distribution** Bernoulli Trials and the Bernoulli Distribution, Binomial Distribution, Mean and Variance of the Binomial Distribution, Application of the Binomial Distribution, the Geometric Distribution, Development from a Poisson, process, Mean and Variance of the Poisson Distribution. **Continuous Distributions:** Uniform Distribution, Mean and Variance of the Uniform Distribution, Distribution, Mean and variance of the Uniform Distribution, Distribution, Mean and Variance of the Exponential Distribution , The Gamma Distribution, The Weibull, Relay.
5	**Recurrent problems**; Manipulation of sums; Number theory; Special numbers; Generating functions. Recursive definition and structural induction, state machines and invariants, recurrences; generating functions.
6	**Discrete Probability Theory** Elementary graph theory, integer congruences, asymptotic notation and growth of functions, permutations and combinations, counting principles, discrete probability.
7	**Stochastic Processes and Queueing** Discrete-Time Markov Chains, Classification of States and Chains Continuous-Time Markov Chains, The Birth-Death Process in Queueing. Considerations in Queueing Models, Basic Single-Server Model with Constant Rates, Single Server with Limited Queue Length, Multiple Servers with an Unlimited Queue, Other Queueing Models.

Recommended Books					
1.	Probability and statistics in engineering	William W Hines, Douglas C Montgomery, David M Goldman Connie M Borror	4th	John Wiley & Sons	2008
2.	Comprehensive mathematics for computer scientists	Guerino Mazzola, Gérard Milmeister and, Jody Weissmann	1st	Springer	2006

3.	Probability and Statistics with Reliability, Queuing and Computer Science Applications	Kishor S. Trivedi	2nd	Wiley	2001

Course code	**:**	CSE-253	**Credit**	**:**	3.0
Title	**:**	Digital Logic Design	**Prerequisite**	**:**	N/A
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Digital logic is the representation of signals and sequences of a digital circuit through numbers. It is used to create circuits and logic gates, as well as to check computer chips. It is the basis for digital computing and provides a fundamental understanding on how circuits and hardware communicate within a computer. Knowledge of digital logic lends itself to many different computer technology design and engineering professions.

Digital logic design techniques form the basis of all digital integrated circuits. Understanding the methods and components are critical both to hardware designers but also software developers who will utilize these hardware components. It is important to know the principles of digital information representation and presents the common components and design methodologies needed to design more advanced systems.

Course Objectives

- Introduce the concept of digital and binary systems.
- Introduce the principles and methodology of digital logic design at the gate and switch level, including both combinational and sequential logic elements.
- Discuss the manipulation and design combination of operators to form higher level functions (multiplexer, counter) and memory element (flip-flop).

Students Learning Outcomes

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

- Represent numbers and perform arithmetic in bases 2, 8, 10, and 16.
- Encode symbols and numbers in binary codes.
- Add and subtract using 2’s complement code.
- Evaluate and simplify logical functions using Boolean algebra.
- Represent logical functions in Canonical form.
- Analyze and design combinatorial circuits.
- Simplify combinatorial circuits using Karnaugh maps.
- Implement functions with NAND-NAND and NOR-NOR logic.

- Analyze and design modular combinatorial logic circuits containing decoders, multiplexers, demultiplexers, 7-segments display decoders and adders.
- Use the concepts of state and state transition for analysis and design of sequential circuits.
- Use the functionality of flip-flops for analysis and design of sequential circuits.
- Introduce computational problem-solving technique.

#	Title and Descriptions
1	**Introduction** Binary digits, logical levels, digital waveforms and timing diagram.
2	**Number System** Binary, octal & hexadecimal; Addition, subtraction, multiplication and division; Codes: BCD, gray codes; error detecting codes and error correcting codes.
3	**Logic Circuits** Gates; Boolean Algebra; De Morgan's theorem; Sum of products and product of sums; Mapping technique; Karnaughmap; Minimization of logic circuits.
4	**Combinational Circuits** Half and full Adders, Subtractor, Encoders and decoders; Comparators; Parity generator; Multiplexers; Demultiplexers.
5	**Sequential Circuits** S-R, J-K, D and T Flip-flops and Latches; Register; Asynchronous and synchronous counter, Counter applications, Memory.
6	**Converters** Analog to Digital (A/D) and Digital to Analog (D/A) conversion techniques.

Recommended Books					
1.	Digital Fundamentals, PHI, 2006.	Thomas L. Floyd		Prentice Hall	2010
2.	Logic and Computer Design Fundamentals	Mano and Kime	4th	Prentice Hall	2008
3.	Digital Systems, Pearson .	R. J. Tocci, N. S. Widmer and G. L. Moss	5th	Prentice Hall	2010

Course code	**:**	CSE-254	**Credit**	**:**	1
Title	**:**	Digital Logic Design Laboratory	**Prerequisite**	**:**	N/A

Type	**:**	Laboratory Work	**Contact hours**	**:**	[26/52]

Rationale
This

Course Objectives
● Introduce the fundamentals of digital logic design through the use of a large number of design problems.
● Describe the relationship between abstract logic characterizations and practical electrical implementations.

Students Learning Outcomes
After successful completion of this course, students should be able to:
● Understand the basic software tools for the design and implementation of digital circuits and systems.
● Analyze the operation of a flip-flop and examine relevant timing diagrams.
● Analyze the operation of counters and shift registers.
● Design and operate practical digital logic circuits.
● Familiarize with fundamental principles of digital design.
● Design classical hardware design for both combinational and sequential logic circuits.

Lab Course Description	
Exp. #	**Title**
1	Verification of basic logic gates. Verify the universality of NAND and NOR gates.
2	Verification of Boolean laws and rules. Implementation of some Boolean expressions.
3	Implementing Boolean expression using only NAND or NOR gates
4	Design and implementation of half adder and full adder circuits.
5	Design and implementation of decoder and encoder circuits.
6	Design and implementation of comparator and code converter circuits.
7	Design and implementation of multiplexer and demultiplexer circuits.
8	Design and implementation of flip-flop circuits
9	Design and implementation of asynchronous and synchronous counter circuits
10	Design and implementation of shift register circuits

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*

Digital Logic Design Trainer Board	VHDL

Course code	**:**	**CSE-255**	**Credit**	**:**	**3.0**
Title	:	**Database Systems**	**Prerequisite**	:	
Type	:	**Theory**	**Contact hours**	:	**39**

Course Objectives

- To provide general concepts of database management systems.
- To give basic knowledge of designing a database.
- To introduce students with database security.
- To prepare students for facing future challenges of database.

Students Learning Outcomes

At the end of the course the students will be able to

- Learn primary concept of database systems.
- Analyze and Design data model.
- Implement database in MySQL/Oracle etc.
- Get fundamental concept of views and authorization.
- Get some basic concepts of advanced databases.

Course Description

#	Descriptions
1	**Introduction** Database system concept; Purpose of database system; View of data; Data models; Conventional file processing; Transaction management; Storage management; Database administrator.
2	**Database Model** Entity-relationship model; Relational model, Network model; Hierarchical model, Database languages, Relational algebra, Integrity constraint, Generalization and Specialization, Developing an ER Diagram.
3	**Structured Query Language** Basic Structure of SQL, String operations, Different set operations, **Aggregate functions**, Handling NULL values, **Nested Subqueries**, View definition, Modification of the Database: Deletion, Insertion and Update operations, Domain Types in SQL, Alteration of Table Structure.
4	**Database Design** Functional dependencies and normal forms; Object-oriented databases; Distributed database; multimedia database, object-relational database, Intelligent database.
5	**File System Structure & Data Warehouse** File organization and retrieval; File indexing; Hashing. Basic concepts of data warehouse and data mart.
6	**Transactions** Introduction to transaction, ACID Properties, Transaction State, Schedule, Conflict Serializability and View Serializability.

7	**OLTP and NoSQL Systems** Basic Concepts of OLAP, Comparison between OLAP and OLTP, Introduction to NoSQL Systems.

Recommended Books				
1.	Database System Concepts	Abraham Silberschatz Professor, Henry F. Korth , S. Sudarshan	6th	McGraw-Hill Education
2.	Database Systems: Introduction to Databases and Data Warehouses	Nenad Jukic, Susan Vrbsky , Svetlozar Nestorov	1st	Prospect Press
3.	Jump Start MySQL: Master the Database That Powers the Web	Timothy Boronczyk	1st	SitePoint
4.	Oracle Database 12c Hands-On SQL and PL/SQL	Satish Asnani (Author)	2nd	Prentice-Hall

---

Course code	**:**	CSE-256	**Credit**	**:**	1
Title	:	Database Systems Laboratory	**Prerequisite**	:	[Prerequisite]
Type	:	Laboratory Work	**Contact hours**	:	26

Lab Objectives

- To train students to use DBMSs (e.g., MySQL, Oracle, etc.)
- To give practical experience in retrieving information from a database system efficiently and effectively.
- To develop ability to design, develop/create, and manipulate a relational database using a DBMS.

Lab Outcome

At the end of the course the students will be able to:

- Design and implement a database schema and populate the database.
- Formulate queries using SQL statements/commands..
- Familiarize with the concepts of database technologies.

Lab Course Description

Exp. #	Title
1	Working with database designing tools (e.g. for E-R diagram drawing.).
2	Setting up a DBMS (MySQL).
3	Creating and populating a database.
4	Retrieving and updating data/information from and to a database.
5	Working with multiple tables in a database.
6	Connection to and Programming (front end and Back end) a database with PHP (also optionally with other tools.).
7	Performing Database Backup and replication.
8	Case studies – Database Design-I
9	Case studies – Database Design-II
10	Case studies – Database Design-Normalization

11	Case studies – Database Design-BCNF
12	Case studies – Database Security
13	Case studies – Database Security
14	Case studies – Database Design- RAID
15	Case studies – Working with Multiple Databases
16	Case studies – Working with Multiple Databases
17	Working with Oracle.
18	Working with MS Access/ any other DBMSs.

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
Computers, etc.	**DMBSs(e.g. MySQL/Oracle/MS Access etc.)**

Course code	**:**	CSE-257	**Credit**	:	3.0
Title	**:**	Algorithms II	**Prerequisite**	:	C/C++/Java, Data Structures
Type	**:**	Theory	**Contact hours**	:	39

Course Objectives

Upon completion of this course, students will be able to do the following:

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Students Learning Outcomes

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

- Argue the correctness of algorithms using inductive proofs and invariants.
- Analyze worst-case running times of algorithms using asymptotic analysis.
- Describe the advanced divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.

- Describe the advanced dynamic-programming paradigm and explain when an algorithmic design situation calls for it and analyze them.
- Explain the advanced graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.
- Compare between different data structures. Pick an appropriate data structure for a design situation.

#	Title and Descriptions
1	**Algorithmic Thinking Overview** Statistics: Mean, Median, Variance, Correlation; Probability:Independent event, Mutually Exclusive Event, Not mutually exclusive events, Conditional Probability, Inverse Probability, Expected Value
2	**Branch and Bound** Analyzing Algorithms, Asymptotic notation, functions and running times, Amortized Analysis
3	**Advanced Divide and Conquer** Property, Recurrence ,Solving Recurrence: Subtitution Method, Master Method,Recursion Tree Method, Proof of Master Method; Strassen's Matrix Multiplication; FFT and DFT; Matrix Exponentiation
4	**Advanced Dynamic Programming** Property, Problems: Edit Distance, Subset Sum, Matrix Chain Multiplication, Optimal Binary Search Tree
5	**Advanced Data Structures** Sparse Table; Segment Tree, Trie, Lowest Common Ancestor, Splay Tree, Red Black Tree
6	**Network Flow and Matching** SCC, Articulation Point and Bridge using (Tarjan algorithm), Flow networks, Ford-Fulkerson method, Max Flow Min Cut Problem, Dinic's Algorithm, Maximum Bipartite Matching, Stable Marriage Problem, Weighted Bipartite Matching, Min Cost Max Flow, The naïve string matching algorithm, String Matching with Finite Automata, The Knuth-Morris-Pratt algorithm; The naïve string matching algorithm, String Matching with Finite Automata, The Knuth-Morris-Pratt algorithm
7	**Hashing and others** Direct-address tables, Hash tables, Hash functions, Open addressing; Computational Geometry: Line-segment property and operations, Convex Hull; Game Theory: Nim, Poker Nim, Hackenbush, Approximation algorithms, The Travelling Salesman Problem, The vertex-cover problem, The set-cover problem; P, NP, NP-hard, NP-Complete Problems

Recommended Books

1.	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	3rd Edition	The MIT Press	
2.	Algorithms Unlocked	Thomas H. Cormen	1st Edition	The MIT Press	
3.	Introduction to the Design and Analysis of Algorithms	Anany Levitin	3rd Edition	Pearson	

Course code	**:**	CSE-258	**Credit**	**:**	1
Title	**:**	Algorithms-II Laboratory	**Prerequisite**	**:**	C / C++ / Java
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Lab Objectives

Upon completion of this course, students will be able to do the following:

- Analyze the asymptotic performance of algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Lab Outcome

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

- Analyze worst-case running times of algorithms using asymptotic analysis.
- Implement the advanced divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
- Implement the advanced dynamic-programming paradigm and explain when an algorithmic design situation calls for it and analyze them.
- Implement and compare between different data structures. Pick an appropriate data structure for a design situation.
- Implement the advanced graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.

Exp. #	Title and Description
1	Computational complexity
2	Strassen's Matrix Multiplication, Exponentiation and Chain Multiplication

3	Multiplication using FFT.
4	Edit Distance and Subset Sum
5	Offline RMQ problem
6	Dictionary search using TRIE
7	Lowest Common Ancestor
8	Splay Tree
9	Self-Balancing Binary Search Tree Red Black Tree
10	SCC, Articulation point and Bridge detection using Tarjan algorithm
11	Ford-Fulkerson Algorithm, Maximum Bipartite Matching and Stable Marriage Problem
12	Dinic's Algorithm
13	Weighted Bipartite Matching
14	Knuth-Morris-Pratt algorithm String matching algorithm
15	Hashing for string matching
16	Convex Hull
17	Nim and its variations

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 / InteliJ / Net beans / Eclipse

---

Course code	**:**	CSE-259	**Credit**	**:**	3.0
Title	**:**	Data and Telecommunication	**Prerequisite**	**:**	
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale
Data communication, which is the transmission of digital data through a network or to a device external to the sending device, is the cornerstone of modern telecommunications. Data communication networks can affect businesses by being the foundations for distributed systems in which information system applications are divided among a network of computers. Data communication networks facilitate more efficient use of computers and improve the day-to-day control of a business by providing faster information flow.

Course Objectives

- Introduce the concept of data communication.
- Introduce the digital and analogue representations and channels.
- Describe the notion of Information and their transmission behavior over the communication channel.
- Introduce digital signal transmission and encoding techniques.

Students Learning Outcomes

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

- Understand the fundamental concepts of data communications and networking
- Identify different components and their respective roles in a data communication system.
- Apply the knowledge, concepts and terms related to data communication and networking.
- Explain the role of line codes in a data communications network.
- Describe the features and functions of multiplexing and modulation.

#	Title and Descriptions
1.	**Information and Communication Channel** Basics on Probability, Information Theory, Hartley's and Shannon theorem on information, Self-information, Types of communication channel, Analog and digital communication, Digital signal nomenclature.
2.	**Pulse Code Modulation and Companding** Analog to digital conversion, Sampling of continuous signal, Quantization of sampled signal, Pulse code modulation, Companding principle and laws.
3.	**Pulse Modulation Techniques** Pulse amplitude modulation (PAM), Pulse width modulation (PWM), Pulse position modulation (PPM), Conversion of PWM to PAM, Differential pulse code modulation (DPCM) and Delta modulation.
4.	**Line Coding** Unipolar, Polar, Bipolar line coding, Nonreturn-to-zero, return-to-zero, Manchester coding (Split phase); Scrambling technique: Bipolar with 8 Zeros Substitution (B8ZS), High Density Bipolar 3 zeros (HDB3).
5.	**Multiplexing techniques** Frequency-division multiple access (FDMA), Time-division multiple access (TDMA), Code-division multiple access (CDMA); Digital modulation techniques: Frequency shift keying (FSK), Phase shift keying (PSK)
6.	**Data link layer** Framing: byte stuffing, bit stuffing; Error control: Automatic repeat request (ARQ); Flow control: Stop and wait protocol, Sliding window protocol.

7.	**ATM and Frame Relay** Asynchronous transfer mode (ATM) multiplexing, Architecture of ATM network, Virtual connection, ATM switching, ATM protocol hierarchy, ATM cell format; X.25 Overview, X.25 layers, X.25 call setup, Frame relay: Frame relay devices, Frame relay layers, Address formats, Comparison of X.25 and frame relay.

Recommended Books

	Title	Authors	Edition	Publisher	Year of Publication
1.	Data Communication and Networking	B. Forouzan	5th Edition	McGraw-Hill	2012
2.	Data and Computer Communication	W. Stallings	10th Edition	Prentice Hall	
3.	Data Communications	P. C. Gupta		Prentice Hall of India	
4.	Communication Systems	S. Haykin		LPE	

Course code	**:**	**CSE-260**	**Credit**	**:**	**1**
Title	**:**	Data and Telecommunication Laboratory	**Prerequisite**	**:**	Basic MATLAB Programming
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

Data communication, which is the transmission of digital data through a network or to a device external to the sending device, is the cornerstone of modern telecommunications. Data communication networks can affect businesses by being the foundations for distributed systems in which information system applications are divided among a network of computers. Data communication networks facilitate more efficient use of computers and improve the day-to-day control of a business by providing faster information flow.

Lab Objectives

To understand the differences between analog and digital communication, the importance of modulation and techniques and noises.

Lab Outcome

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

After completion of the course, a student gets clear understanding of digital communication. He is also able to grasp the necessity of different techniques for meaningful data transmission.

Exp. #	Title	Contact Hours
1	AM and FM modulation and demodulation	3
2	Verification of Gibb's phenomenon using rectangular and saw tooth wave	3
3	Quantization technique of an analog signal	3
4	Simulation of binary data communication	3
5	Signal modulation technique: a. Data modulation b. Analog modulation	3
6	Differential pulse code modulation and Delta modulation	3
7	Huffman coding	3
8	Simulation of a digital communication system	3
9	SNR of a communication channel	3

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
Computer systems with i-core processors and 8GB RAMs.	**Matlab**

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

Third Year First Semester

Course code	:	CSE 300	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

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

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

Course code	:	ECO 301	Credit	:	3.0
Title	:	Economics	**Prerequisite**	:	None
Type	:	*Theory*	**Contact hours**	:	39

Rationale
Understanding principles of economics has immense importance for scientific solution of the problems of resource allocation. By conducting this course students will be acquainted with a thorough grounding in the basic principles of economics and an exposure to a range of applications of the theory in real world problems.

Course Objectives
The learning objective of this course is to provide a self-contained introduction to economics principles, develop an understanding of fundamental concepts in micro and macroeconomic analysis, understating cryptocurrency and equip students with a range of appropriate analytical skills including descriptive and graphical methods for solving real world problems.

Students Learning Outcomes

At the end of the course the students will be able to:

- Understand the key ideas that define the economic way of thinking as software engineer.
- Demonstrate substantial knowledge on fundamental economic question of allocating scarce resources, principles of demand, supply, market price and quantity determination.
- Grasp the knowledge of production theory, firm behavior and cost.
- Analyze the measurement of macroeconomic aggregates.
- Understand and apply the basic principles related to industrial economics and resource economics.
- Evaluate various policy decisions related to global pollution.
- Understand the concept of cryptocurrency and its significance.

#	Topics and Descriptions
1.	**Introduction**

	Definition of economics; nature and scope of economics; micro versus macroeconomics; economic good versus free good.
2.	**Demand and Supply** Concepts of demand and supply; law of demand and supply; determinants of demand and supply; movement along demand and supply curves; shifting of demand and supply curves; market demand curve; market equilibrium; shift of equilibrium; consumers' surplus and producers' surplus.
3.	**Economics of Production, Cost and Revenue** Factors of production; production function; total, average and marginal products; stages of production; law of diminishing return; returns to scale; isoquant, isocost line and producers' equilibrium. Concept of cost; Short-run and long-run cost; Fixed and variable cost; Total, average and marginal cost; Shape of and inter-relationship between different types of cost curves; Concept of – total, average and marginal revenue.
4.	**Basic Macro Economic Concepts** GNP, GDP, NNP, NI; circular flow of income; inflation; unemployment; business cycle; fiscal policy; monetary policy.
5.	**Economics of Industry, Regional and Global Pollution** Concept of firm and industry; measuring the size of firms: small, medium and large scale firms; concept of optimum firm; input-output analysis of inter-industry relation, study of major industrial countries: background, effects, current economic situation; industrialization in Bangladesh; government measurement; Types of pollutant; sources of pollution; pollution heaven hypothesis; scope of environmental damage; damage function; environmental quality; sustainable development; risk analysis; pollution reduction versus
6.	**Resource and Environmental Economics** Definition; types of resources; relation between economics and ecology; categories of resources on the basis of degree of economic importance and discovery; theory of optimal harvest of renewable resources; forest and energy resource of Bangladesh; Concepts of environmental economics; Material Balance Model; role of economics in environmental managements; cost-benefit analysis in environmental decision making.
7.	**Cryptocurrency** Introduction to Crypto and Cryptocurrencies, List of Cryptocurrencies, How Bitcoin Achieves Decentralization, Mechanics of Bitcoin, How to Store and Use Bitcoins, Bitcoin Mining, Impact of cryptocurrency as an alternative monetary system.

Recommended Books

1.	Fundamentals of Economics	William Boyes, Michael Melvin	6th	Cengage Learning	2012
2.	Economics For Dummies	Sean Masaki Flynn	2nd	For Dummies	2011

3.	CRYPTOCURRENCY: The Complete Basics Guide For Beginners. Bitcoin, Ethereum, Litecoin and Altcoins, Trading and Investing, Mining, Secure and Storing, ICO and Future of Blockchain and Cryptocurrencies	Michael Horsley	1st	CreateSpace Independent Publishing Platform	2017
4.	Mastering Bitcoin: Programming the Open Blockchain	Andreas M. Antonopoulos	2nd	O'Reilly Media	2017

Course code	:	CSE 303	Credit	:	3.0
Title	:	Computer Graphics	**Prerequisite**	:	CSE-105
Type	:	*Theory*	**Contact hours**	:	39

Rationale
Computer graphics is one of the most exciting and rapidly growing computer fields and has many applications, including user interfaces, data visualization, computer-aided design, motion pictures and image processing. This unit concentrates on fundamentals of computer graphics and addresses the knowledge and skills in computer graphics development which are essential for computing professionals.

Course Objectives
The goal of this course is to provide an introduction to the theory and practice of computer graphics and data visualization. The course will assume a good background in programming in C or C++ and a background in mathematics including familiarity with the theory and use of coordinate geometry and of linear algebra such as matrix multiplication. There is a choice of both breadth and depth in the intertwined topics of graphic, computational geometry, geometric modeling and data visualization.

Students Learning Outcomes
After completing this course students will be able to: ● Understand fundamental theories and algorithms in computer graphics such as geometric projections and transformations, view and projection models, and different lightning models and algorithms for rendering polygon-based objects ● Derive and apply geometric view and projection models and transformations of homogeneous coordinates in computer graphics ● Derive and apply basic rendering techniques and algorithms in polygon-oriented computer graphics such as lightning models, line and polygon cutting algorithms

- Describe and relate various visual effects such as antialiasing, texture mapping.

#	Topics and Descriptions
1.	**Graphics Overview** Display devices, Raster refresh graphics display, use of frame buffer and look-up table. Device coordinate, normalized device coordinate and world coordinate system
2.	**Graphic I-O Devices and Scan Conversion** Raster scan graphics: Bresenham's line and circle generation algorithms, character generation, half-toning, antialiasing, Random scan displays, raster refresh graphics displays, interactive devices, logical functioning of graphic I-O devices, output devices. Scan-conversion of a point, a line, a circle, an ellipse, arcs and sectors, a polygon, a character, region fillings, aliasing effects, anti-aliasing, image compression, recursively defined drawings.
3.	**Graphics Programming and Computer Animation** Overview of tools and techniques of graphics programming using OpenGL, MATLAB, C/C++, JAVA. The nature of computer animation, simulation, kinematics, parametrics, dynamics, metamorphosis, displacement, animation, problems in animation, techniques of animation.
4.	**Transformations and Projections** 2D and 3D Transformations, geometric transformations, coordinate transformations, composite transformations, instance transformations, shearing transformations. Projections: taxonomy of projection, perspective projection, parallel projection
5.	**Ray Tracing and 3D Viewing and Representation** The pinhole camera, recursive ray-tracer, parametric vector representation of a ray, ray-surface intersection, execution efficiency, anti-aliasing, additional visual effects. Simple geometric forms, wireframe models, curved surfaces, curve design, polynomial basis functions, the problem of interpolation and approximation, curved surface design, transforming curves and surfaces, quadratic surfaces, terrain generation, fractal geometry methods
6.	**Hidden Surface** Depth comparisons, Z-buffer algorithm, back-face removal, the Painter's algorithm, scan-line algorithm, subdivision algorithm, hidden-line elimination, rendering of mathematical surfaces, Warnock's algorithm, Weiler-Atherton algorithm.
7.	**Color Models, Shading and Rendering** Light models, shading, interpolation techniques: constant, Gouraud and Phog, ray tracing, computer ergonomics, information structure, introduction to graphics kernel system.

Recommended Books

1.	Interactive Computer Graphics	Ed. Angel, Dave Shreiner	7th	Pearson	2014
2.	Graphics and Visualization: Principles & Algorithms	T. Theoharis, G.Papaioannou, N. Platis, N. M. Patrikalakis	1st	A K Peters/CRC	2007
3.	Schaum's Outline of Computer Graphics	Zhigang Xian, Roy A. Plastock	2nd	The McGraw-Hill Companies	2015
4.	Computer Graphics: Theory and Practice with OpenGL	Zhigang Xian	1st	CreateSpace Independent	2018

Course code	:	CSE-304	Credit	:	1
Title	:	Computer Graphics Laboratory	**Prerequisite**	:	CSE-105
Type	:	Laboratory Work	**Contact hours**	:	26

Rationale

Computer graphics is one of the most exciting and rapidly growing computer fields and has many applications, including user interfaces, data visualization, computer-aided design, motion pictures and image processing. This unit concentrates on the hands on experience of the fundamentals of computer graphics which are essential for computing professionals.

Lab Objectives

- To develop an interactive programs for 2D and 3D transformation
- To develop algorithms for scan conversion of basic drawing
- To apply OpenGL for ray tracing and rendering

Lab Outcome

At the end of the course the students will be able to:

- Understand graphics programming.
- Be exposed to analyzing of 2D graphical scenes using open graphics library suits.
- Be familiar with image manipulation, enhancement.

- Study to create simple animations.

Lab Course Description		
Exp. #	**Title**	**Contact Hours**
1.	Implementation of Algorithms for drawing 2D Primitives – Line	3
2.	(DDA, Bresenham) – all slopes Circle (Midpoint)	3
3.	Implementation of Line, Circle and ellipse Attributes.	3
4.	2D Geometric transformations –Translation Rotation Scaling	3
5.	Creating two dimensional objects	3
6.	3D Transformations – Translation, Rotation, Scaling	3
7.	Coloring the pictures	3
8.	Curve generation	3
9.	Image Editing and Manipulation – Basic Operations on image using any image editing software, Creating gif animated images, Image optimization.	2

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
Standalone desktops – 30 Nos. or Server supporting 30 terminals or more, 2GB HDD or greater, 4GB of RAM for animation	C, C++, Java, OpenGL, LuxRender, 3ds Max, Blender, Carrara, Cinema 4D, DAZ Studio, Maya, Poser, SketchUp, XSI, Microsoft windows 8/10(64/32 bit) Adobe Flash Professional version 6

Course code	:	CSE-305	Credit	:	3.0
Title	**:**	Computational Geometry	**Prerequisite**	**:**	Data Structure and Algorithm Analysis
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale
Computational Geometry rests on a solid foundation built in the last three decades, which has given the community a firm grasp of the fundamental issues arising in questions of visibility, proximity, intersection, and multidimensional searching, as well as shape analysis, reconstruction, and modeling. This unit concentrates on the fundamental solutions requiring meeting a number of challenges, including how to reason about shapes at different

levels of resolution and how to cope with high dimensions.

Course Objectives

The primary goal is to develop algorithms and data structures for solving problems stated in terms of basic geometrical objects: points, line segments, polygons, polyhedra etc. This course will introduce basic tools and algorithms and address fundamental problems such as convex hulls, polygon triangulation, range search, Voronoi diagrams, Delaunay triangulations etc.

Students Learning Outcomes

- After completing this course the students will be able to:
  - Design algorithms for simple geometrical problems
  - Apply geometric techniques to real-world problems in graphics
  - Solve linear programs geometrically

#	Topics and Descriptions
1.	**Geometric Fundamentals** Introduction, Algorithm and complexity of fundamental geometric objects, geometric data structures, Geometric Preliminaries, mathematical models of computation and lower bounds.
2.	**Geometric Searching and Data Structures** Point-location, Orthogonal Range Searching, interval and segment trees, data structures for nearest neighbors, range searching, multi-dimensional search trees(k-d trees).
3.	**Convex Hulls** Preliminaries, Problem Statement and Lower Bounds, Convex Hull Algorithms in the Plane, Graham's Scan, Jarvis's March, QUICKHULL techniques, Dynamic Convex Hull, Convex Hull in 3D
4.	**Triangulations** Polygon triangulations and art gallery theorem, Planar Triangulations, Greedy Triangulations, Partitioning a Polygon into Monotone Pieces, Triangulating a Monotone Polygon, Delaunay Triangulation
5.	**Intersections** Intersections of linear objects: two lines, two line segments, n line segments, two convex polygons, general polygons, etc, Application Areas, Planar Applications: Intersection of Convex Polygons, Star-shaped Polygons; 3D Applications: Intersection of 3D Convex Polyhedra; Intersection of Half-spaces

6.	**Linear Programming** Half-Plane Intersection, Incremental Linear Programming, Randomized Linear Programming, Unbounded Linear Programs, The Smallest Enclosing Disk Problem
7.	**Proximity Problem** A Collection of Problems, A Computational Prototype: Element Uniqueness, Lower Bounds, The Closest-Pair Problem: A Divide-and-Conquer Approach, The Voronoi Diagram, Proximity Problems Solved by the Voronoi Diagram

Recommended Books

1.	Computational Geometry: Algorithms and Applications	Mark de Berg, Mark Overmars, Otfried Schwarzkopf, M. van Kreveld	3rd	Springer	2011
2.	Computational Geometry in C	Joseph O'Rourke	2nd	Cambridge Uni. Press	1995
3.	Handbook of Discrete and Computational Geometry	Jacob E. Goodman, Joseph O'Rourke	1st	CRC Press	2017
4.	Discrete and Computational Geometry	Satyan L. Devadoss and Joseph O'Rourke	---	Princeton Uni. Press	2012

Course code	:	CSE 307	Credit	:	3.0
Title	:	Computer Architecture and Organization	**Prerequisite**	:	None
Type	:	*Theory*	**Contact hours**	:	39

Rationale

Computer systems play an integral role in all facets of the engineering profession. Systems users are always in need of faster, more powerful, yet cheaper computer systems. This course provides an understanding of the processor-level components of computer systems, their design and operation, and their impact on the overall performance of the systems.

Course Objectives

The learning objective of this course is to provide a clear idea about computer organization and architecture: instruction formats and construction; addressing modes; memory hierarchy (cache, main memory and secondary memory) operation and performance; simple pipelines; basic performance analysis; simple OS functions, particularly as they relate to hardware; virtual memory; computer I/O concepts, including interrupt and DMA

mechanisms; inter-computer communication concepts

Students Learning Outcomes

At the end of the course the students will be able to:

- identify the elements of modern instructions sets and explain their impact on processor design,
- identify and explain the function of basic elements of a modern processor, including instruction pipelines,
- explain the function of each element of a memory hierarchy,
- identify and compare different methods for computer I/O,
- compare simple computer architectures and organizations based on established performance metrics.

#	Topics and Descriptions
1.	**Introduction** Instruction codes, formats, cycle, timing etc; Addressing modes; Types of instruction; RISC characteristics; CISC characteristics.
2.	**Central Processor** Instruction Sets & Operands, Fetch-Execute Cycle, Simple Pipelining, Control Unit Operation
3.	**Computer Arithmetic** Different types of data representation; Addition and subtraction; Multiplication algorithms; Division algorithms.
4.	**Memory Organization** Main memory; Auxiliary memory; Associative memory; Cache memory; Virtual memory; Memory management requirements and hardware.
5.	**Input-Output Organization** Input-Output Interfaces; Data transfer, Interrupts; Direct Memory Access (DMA); Input-output channel.
6.	**Fundamentals of parallel processing** Parallel processing; Pipelining; Multiprocessors; Array processor, Bit-slice processor Interconnection structures, Multicore Architecture.
7.	**Data-Level Parallelism** SISD, MIMD, SIMD, SPMD, Vector Architecture, Graphics Processing Units

Recommended Books

1.	Computer Architecture: A Quantitative Approach	John L. Hennessy, David A. Patterson	6th	Morgan Kaufmann	2016
2.	Computer Organization and Architecture	William Stallings	10th	Pearson	2015
3.	Essentials of Computer Organization and Architecture	Linda Null	5th	Jones & Bartlett Learning	2018
4.	Computer Architecture Organization	John P. Hayes	2nd	McGraw-Hill	2003

Course code	:	CSE 309	Credit	:	3.0
Title	:	Operating Systems	**Prerequisite**	:	None
Type	:	*Theory*	**Contact hours**	:	39

Rationale
The operating system provides an established, convenient, and efficient interface between user programs and the bare hardware of the computer on which they run. This unit focuses on the basic operating system abstractions, mechanisms.

Course Objectives
The goal of this course is to provide an introduction to the internal operation of modern operating systems. Besides, the course will introduce the students to modern operating systems design. Both practical and theoretical aspects of Operating Systems will be studied. In particular, the course will cover processes and threads, mutual exclusion, CPU scheduling, deadlock, memory management, and file systems.

Students Learning Outcomes
After completing the course, students will be able to: ● Understand the main responsibilities of a contemporary operating system (OS),the goals of standardization of OS (and other) interfaces; the structure of operating systems, applications, and the relationship between them. ● Understand operating system design and its impact on application system design and performance ● Achieve the competency of recognizing, analyzing, evaluating operating system features.

#	Topics and Descriptions

1.	**Overview** Overview of operating systems, Evolution of OS, functionalities and characteristics of OS.
2.	**Hardware concepts** Hardware concepts related to OS, CPU states, I/O channels, memory hierarchy, microprogramming
3.	**Process** The concept of a process, operations on processes, process states, concurrent processes, process control block, process context, Process control, Interrupt processing, operating system organization, OS kernel FLIH, dispatcher. UNIX process control and management, PCB, signals, forks and pipes
4.	**CPU Scheduling** Job and processor scheduling, scheduling algorithms, process hierarchies.
5.	**Concurrency and Synchronization** Problems of concurrent processes, critical sections, mutual exclusion, synchronization, deadlock, Mutual exclusion, process co-operation, producer and consumer processes. Semaphores: definition, init, wait, signal operations. Use of semaphores to implement mutex, process synchronization etc., implementation of semaphores.
6.	**Memory and Storage Management** Principles, requirements and design of memory management system, program loading and linking. virtual memory, page table, translation lookaside buffer, segmentation, software implementation, load control, paging and segmentation, address mapping, virtual storage management, page replacement strategies, storage allocation.
7.	**I/O and File Management.** Direct memory access. Design issues. I/O buffering. Disk I/O. Disk cache. Example systems. File management systems, File organization and access, file directories. Sharing of files. Record blocking. Secondary storage management, file systems protection and security; design and implementation methodology. Example systems.

Recommended Books

1.	Operating System Concepts	A. Silberschatz, P. B. Galvin, Greg Gagne	10th	John Wiley & Sons.	2017
2.	Modern Operating Systems	A. S. Tanenbaum, Herbert Bos	4th	Pearson	2014
3.	Operating Systems, Internal and design principles	William Stallings	9th	Pearson	2017
4.	Operating Systems	Harvey M. Deitel, Paul	3rd	Pearson	2003

		J. Deitel, David R. Choffnes			

Course code	:	CSE-310	Credit	:	1.0
Title	**:**	Operating Systems Laboratory	**Prerequisite**	**:**	
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale
The operating system provides an established, convenient, and efficient interface between user programs and the bare hardware of the computer on which they run. This unit focuses on the basic operating system abstractions, mechanisms, and their implementations.

Lab Objectives
● Learn shell programming and the use of filters in the UNIX environment. ● Be exposed to programming in C using system calls. ● Learn to use the file system related system calls. ● Be familiar with implementation of CPU Scheduling Algorithms, page replacement algorithms and Deadlock avoidance.

Lab Outcome
After completing this Laboratory course, the students will be able to: ● Compare the performance of various CPU Scheduling Algorithms. ● Implement deadlock avoidance, and Detection Algorithms. ● Critically analyze the performance of the various page replacement algorithms.

Lab Course Description

Exp. #	**Title**	**Contact Hours**
1.	Basics of UNIX commands.	3

2.	Shell programming	3
3.	Implementation of CPU scheduling. a) Round Robin b) SJF c) FCFS d) Priority	3
4.	Implement all file allocation strategies	3
5.	Implement Semaphores for handling process synchronization	3
6.	Implement Bankers algorithm for Dead Lock Avoidance and detection	3
7.	Implement the all page replacement algorithms a) FIFO b) LRU c) LFU	3
8.	Implement Paging Technique of memory management	3
9.	Implement Threading & Synchronization Applications	2

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
High configuration PCs equipped with required software	C, C++, Java, Equivalent complier, Microsoft windows 8/10(64/32 bit), Linux operating system, Nachos

---

Course code	:	CSE-312	Credit	:	2.0
Title	**:**	Web Design and Programming Laboratory-I (PHP/C#)	**Prerequisite**	**:**	C/C++
Type	**:**	Laboratory Work	**Contact hours**	**:**	52

Rationale

Web Design and Programming involve the standards for building and Rendering Web pages, including HTML, CSS, SVG, device APIs, and other technologies for Web Applications (“WebApps). The Web Design and Programming Lab-I curriculum is an introduction to the design, creation, and maintenance of web pages and websites which will help the students to evaluate website quality, learn how to create and maintain quality web pages, learn about web design standards and why they're important, and learn to create and manipulate images.

Lab Objectives

This course is an introduction to programming for the World Wide Web. We will cover all the major pieces of how websites work. This will include the relationship between clients and servers, how web pages are constructed, and how the internet works. We’ll examine several technologies in depth:

- HyperText Markup Language (HTML) for authoring web pages

- Cascading Style Sheets (CSS) for styling web pages
- JavaScript for creating interactive web pages
- Asynchronous JavaScript and XML (Ajax) for enhanced web interaction and applications
- JSON for transferring data
- PHP/ C# for generating dynamic pages on a web server
- Structure Query Language (SQL) for interacting with databases

Lab Outcome

After completing this Laboratory course, the students will have:

- Detail knowledge of the relationship between client and server and client-site and server-side programming.
- Practical knowledge of languages of HTML, CSS, Java Scripts, Ajax, and PHP/C# language
- Hands-on experience of design and development of web application.

Lab Course Description

Exp. #	**Title**	**Contact Hours**
1.	Creating form with HTML elements-I	3
2.	Creating form with HTML elements-II	3
3.	Web page design with CSS-I	3
4.	Web page design with CSS-II	3
5.	Java scripts programming-I	3
6.	Java Script Programming-II	3
7.	Java Script Programming-III	3
8.	Asynchronous Programming with AJAX	3
9.	Programming with PHP-I	2
10.	Programming with PHP-II	3
11.	Web Database connectivity and data manipulation	3
12.	AngularJS/ NodeJS/ ExpressJS with PHP	3
13.	Understanding Visual Studio and C#	3
14.	Programming C#.NET with Visual Studio-I	3

15.	Programming C#.NET with Visual Studio-II	3
16.	Programming C#.NET with Visual Studio-III	3
17.	Programming C#.NET with Visual Studio-IV	3
18.	Programming C#.NET with Visual Studio-V	2

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
High configuration PCs equipped with required software	phpStorm and Visual Studio 2013 or later

---

Course code	:	CSE 314	Credit	:	2.0
Title	**:**	OOAD Laboratory	**Prerequisite**	**:**	CSE-162, CSE-212
Type	**:**	Laboratory Work	**Contact hours**	**:**	52

Rationale

Object-oriented analysis and design (OOAD) is a popular technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modeling throughout the development life cycles to foster better stakeholder communication and product quality. This unit focuses on the basic concepts of OOAD which will help to create a model of the system's functional requirements that is independent of implementation constraints.

Lab Objectives

- To teach the students a solid foundation on object-oriented principles.
- To teach the student the essential and fundamental aspects of object-oriented analysis and design, in terms of “how to use” it for the purpose of specifying and developing software.
- Explain the importance of good requirement gathering and risk management
- Introduce the fundamental principles through advanced concepts of analysis and design using UML.
- Explain the benefits and the risks of using UML.

Lab Outcome

After completing this Laboratory course, the students will be able to:

- Show the importance of systems analysis and design in solving complex problems.
- Show how the object-oriented approach differs from the traditional approach to systems analysis and design.
- Construct various UML models (including use case diagrams, class diagrams, interaction diagrams, state chart diagrams, activity diagrams, and implementation diagrams) using the appropriate notation.
- Recognize the difference between various object relationships: inheritance, association, whole-part, and dependency relationships.
- Show the role and function of each UML model in developing object-oriented software

Lab Course Description		
Exp. #	**Title**	**Contact Hours**
1.	Revisit OOP basic concept	3
2.	To develop a problem statement.	3
3.	Develop an IEEE standard SRS document. Also develop risk management and project plan (Gantt chart). / Preliminary Project Plan	3
4.	Identify Use Cases and develop the Use Case model.	3
5.	Interim Project Phase Presentation I	
6.	Identify the business activities and develop an UML Activity diagram.	3
7.	Identity the conceptual classes and develop a domain model with UML Class diagram.	3
8.	Interim Project Phase Presentation II	
9.	Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.	3
10.	Draw the State Chart diagram.	3
11.	Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.	3
12.	Interim Project Phase Presentation III	
13.	Implement the Technical services layer.	2
14.	Implement the Domain objects layer.	3

15.	Implement the User Interface layer.	3
16.	Draw Component and Deployment diagrams.	3
17.	Demonstration of Final Project	3
18.	Demonstration of Final Project	3

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
High configuration PCs equipped with required software	Umbrello, ArgoUML, IntelliJ IDEA, IBM Rational Rose XDE.

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

Third Year Second Semester

Course code	**:**	CSE 350	**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

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

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

Course code	:	BIS 351	**Credit**	:	3.0
Title	:	Management and Accounting	**Prerequisite**	:	None
Type	:	*Theory*	**Contact hours**	:	39

Rationale

The course gives an introduction to management and accounting. It enables students to develop knowledge and skills to plan, control and manage financial affairs of individuals, communities, and businesses.

Course Objectives

- To learn about basic concepts and principles of business and accounting
- To introduce the role of accounting and financial reporting in market economy
- To develop the skill of evaluating the performance of business organization

Students Learning Outcomes

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

- Understand the accounting rules and regulations required for business enterprises
- Prepare financial statements in accordance with Generally Accepted Accounting Principles
- Explain the concepts and procedures of financial reporting, including income statement, statement of retained earnings, balance sheet, and statement of cash flows
- Apply management accounting ideas and experiences in making decisions in businesses

Course Description

#	Title and Descriptions
1	Objectives and importance of accounting; Accounting as an information system; computerized system and applications in accounting;
2	Recording system: double entry mechanism; Accounts and their classification; Accounting equation;
3	Accounting cycle: journal, ledger, trial balance; Preparation of financial statements considering adjusting and closing entries; Accounting concepts (principles) and conventions.
4	Financial statement analysis and interpretation: ratio analysis.
5	Cost concepts and classification; Overhead cost: meaning and classification; Distribution of overhead cost; Overhead recovery method/rate;
6	Job order costing: preparation of job cost sheet and quotation price;
7	Inventory valuation: absorption costing and marginal/variable costing technique; Cost-Volume-Profit analysis: meaning, breakeven analysis, contribution margin approach, sensitivity analysis.
8	Short-term investment decisions: relevant and differential cost analysis. Long-term investment decisions: capital budgeting, various techniques of evaluation of capital investments

Recommended Books					
1.	Accounting Principles	J. J. Weygandt, D. E. Kieso	12th	John Wiley & Sons	2015
2.	Fundamentals of Accounting of Course	R.E. Ross Claudia, B Gilbertson, Mark W. Lehman, O.D. Manson	7th	PHI	2009
3.	Advanced accountancy	H. Chakrabarty	3rd	Oxford University Press	1978
4	FINANCIAL ACCOUNTING: A MANAGERIAL PERSPECTIVE	R. NARAYANASWAMY	4th	PHI	2011

Course code	**:**	CSE 353	**Credit**	**:**	3.0
Title	:	Human Computer Interaction	**Prerequisite**	:	Computer Peripherals, Computer Programming
Type	:	*Theory*	**Contact hours**	:	39

Rationale

The course gives an introduction to Human Computer Interaction design principles. Students will get basic knowledge about how to develop methodologies, explore various interaction paradigms, evaluate and implement interactive hardware and software computing systems for supporting human–computer interaction in a novel way.

Course Objectives

Upon successful completion of this course the student will be able to understand:

- The definitions and foundations of the HCI domain.
- Strategies to study a human user population.
- User-centered techniques for designing interfaces and interactive solutions.
- Evaluation methods, quality factors, and data analysis techniques.
- The research frontiers of HCI including universal design, responsive design and pervasive computing.

Students Learning Outcomes

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

- Understand the basics of human and computational abilities and limitations.
- Understand basic theories, tools and techniques in HCI.
- Understand the fundamental aspects of designing and evaluating interfaces

Course Description

#	**Title and Descriptions**
1	**Human and computer fundamentals** Human memory, Thinking: Reasoning and Problem solving, Psychology and the design of interactive system, Text entry devices, Display devices, Devices for virtual reality and 3D interaction, Processing and Networks

2	**User Studies and UI design** Data collection, Interviews & Surveys, Observation, Diaries, Focus Groups, Ethnographic studies, Personas, Planning and conducting, Task analysis, requirements elicitation, mixed-fidelity prototyping, wireframes, storyboards, sketches & mockups
3	**Interaction Design** Frameworks and HCI, Ergonomics, Interaction styles: Navigation in 3D and 2D, Paradigms, WIMP, multimodality, BCI, cross-device interaction
4	**HCI in Software process** Usability engineering, Interactive design and prototyping, Design rationale, Usability support principles, Golden rules and heuristics, HCI patterns
5	**Evaluation and support system** Expert evaluation models: analytic methods, review methods, model-based methods, Approaches involving user: experimental methods, observation methods, query method, User support styles: command-based methods, context-sensitive help, wizards and assistants, adaptive help
6	**Cognitive models** Cognitive architecture, Hierarchical models, Linguistic models, Physical and device models, Task analysis
7	**Communication and collaboration model** Face-to-face communication, Conversation, Text-based communication, Group working, Speech act theory, Dialog design notation, Diagrammatic notations, Dialog semantics
8	**Ubiquitous computing** Ubicomp, Ubiquitous computing application research, Virtual reality, Augmented reality, Information visualization
9	**Multimedia and Presentation** Hypertext, multimedia, world wide web, rich content, Animation, Static and dynamic web content
10	**HCI over various periods** Past and future of HCI: the past, present and future, perceptual interfaces, context-awareness and perception

Recommended Books

1.	Human-Computer Interaction	Alan Dix, Janet E. Finlay, Gregory D. Abowd, Russell Beale	Third	Prentice-Hall	2004
2.	Human-Computer Interaction Handbook	Andrew Sears, Julie A. Jacko	Third	CRC Press	2009
3.	Research Methods in Human Computer Interaction	Jonathan Lazar, Jinjuan Feng and Harry Hochheiser	Second	Wiley Publishing	2010

Course code	**:**	CSE 355	**Credit**	**:**	3.0
Title	:	Introduction to Bioinformatics	**Prerequisite**	:	None
Type	:	*Theory*	**Contact hours**	:	39

Rationale

This course is designed to enable students to apply information technology and computational intelligence in studying the living things specifically at the molecular level. Students can develop efficient methods and software tools for understanding the relationships among the members of large biological data sets, such as sequence alignment, gene finding, genome assembly, drug discovery, protein structure alignment, prediction of gene expression, genome-wide association studies, the modeling of evolution and cell division etc.

Course Objectives

- To give a broad outline of bioinformatics and its scope and familiarize with the terminology
- To grow elementary understanding of the important notions of bioinformatics
- To develop the skill to utilize the tools and methods used in biological datasets
- To promote the programming skill to implement the basic algorithms in bioinformatics
- To grow the skill to deal with the multidisciplinary problems

Students Learning Outcomes

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

- Explain and comprehend bioinformatics, genomics, proteomics, and phylogenetic concepts
- Work with and use commonly available tools and techniques efficiently on biological datasets (offline/online)
- Develop solutions to bioinformatics problem and write programs to implement basic algorithms

Course Description

#	Title and Descriptions
1	**Review of molecular biology** Genes and DNA, RNA, Amino acids, Protein, Enzymes, genetic code, etc.
2	**Information organization and sequence databases, molecular evolution** Public databases, primary and secondary databases, natural selection, mechanisms of mutations, etc.
3	**Substitution matrices & pairwise sequence alignment** Introduction, Identity substitution matrices, PAM, BLOSUM, Sliding window, dot plot The Needleman and Wunsch Algorithm, The Smith-Waterman Algorithm, FASTA, BLAST, etc.
4	**Brief introduction to Perl & MySQL** **a. Programming** *Perl*: Downloading and installing Perl, Basic Perl syntax and logic, References, Subroutines and modules, Regular expressions, File handling and directory operations, Error handling, etc. **a. Database/DBMS** *MySQL*: Installing and configuring a MySQL server, Relational Database Design, Database access using SQL, etc.
5	**Multiple Sequence Alignment (MSA), & Protein structure prediction**

	Multiple Sequence Alignment, Experimental methods of protein structure determination, Protein folding, Protein structure prediction methods, etc.
6	**Phylogenetics & Hidden Markov Model (HMM)** Phylogeny and phylogenetics, Two classes of tree-generation methods, Application of phylogenetics to studies of the origin of modern humans, Phylogenetic Tree of Life, Introduction to Hidden Markov Model, etc.
7	**Genomics, Transcript and protein expression analysis** DNA sequencing-dideoxy method, Polymerase chain reaction (PCR), The human genome, Basic principles of gene expression, Proteome, etc.

Recommended Books

1.	Concepts in Bioinformatics and Genomics	Jamil Momand (Author), Alison McCurdy (Author), Silvia Heubach (Contributor), Nancy Warter-Perez (Contributor)	1th	Oxford University Press	2016
2.	Fundamental Concepts of Bioinformatics	Dan E. Krane (Author), Michael L. Raymer (Author)	1th	Pearson	2002
3.	Building Bioinformatics Solutions: with Perl, R and MySQL	Conrad Bessant (Author), Darren Oakley (Author), Ian Shadforth (Author)	2nd	Oxford University Press	2014
4.	Molecular Biology: Structure and Dynamics of Genomes and Proteomes	Jordanka Zlatanova (Author), Kensal E. van Holde (Author)	1th	Garland Science	2015

Course code	**:**	CSE 357	**Credit**	**:**	3.0
Title	**:**	Microprocessors	**Prerequisite**	**:**	None
Type	**:**	*Theory*	**Contact hours**	**:**	39

Rationale

This course makes students familiar with the architecture and the instruction set of an Intel microprocessor. Students get the opportunity to study machine addressing, stack operations, subroutines, and programmed and interrupt driven I/O and the basic concepts of machine organization of a modern digital computer.

Course Objectives

- To learn about basic concepts and principles of microprocessor and instruction set architecture
- To study the structure and timing of typical microprocessors, memories, UARTS, timer/counters, serial devices and related devices, MUX and related control structures for building systems
- To become familiar with interrupt programming and hardware/software design tradeoffs
- To introduce various advanced processor architectures such as 80X86, Pentium and Multicore Processors

Students Learning Outcomes

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

- Design, build and program embedded systems including both hardware and software efficiently
- Explain the architecture of different types of microprocessor instructions, the use of procedures and macros, interrupts, the interfacing of various peripheral devices with the microprocessor etc.
- Compare the characteristics of various microprocessors
- Identify and explain the need for advance microprocessors

#	Topics and Descriptions
1.	**Introduction to Microprocessors and Microcomputer-Based Applications** Evolution of Microprocessor, Microprocessor Data Types, Microcomputer Hardware, Input/Output(I/O), Microcomputer System Software and Programming Concepts
2.	**Intel 8086 and its higher versions (80286, 80386)** Introduction, Internal architecture, 8086 Addressing modes, 80286 Memory Management, Protection, Basic 80386 Programming Model, 80286 and 80386 addressing modes
3.	**I/O Interfaces, DMA Controllers and Interrupts** Serial and Parallel Communication Interfaces, Synchronous and Asynchronous Communications, 8251 A Programmable Communication Interface; General Organization of a DMA controller, Organization of an 8237 and its associated logic, Interrupt and Interrupt service routine
4.	**Keyboard and Display** Keyboard design, Display Design, 8279 Keyboard Display Controller
5.	**Semiconductor Memory** General memory Organization and types of memory, General memory Organization, Static RAM and Dynamic RAM, 16K×8 memory module for a maximum mode 8088
6.	**Pentium Microprocessors** Introduction, Architecture, Register sets, Cache, Floating Point Operations, Addressing Modes, Paging, Instruction Set, Opcode, Interrupt, Protected Mode Operations, Hyper-Threading Technology.
7.	**Next Generation Microprocessor** Intel Core Architecture, Intel Dual Core, Core 2 Duo, Core 2 Quad, Core i3, Core i5, Core i7, Mobile Microprocessors, ARM, Helio, Atom.

Recommended Books

1.	Microcomputer Systems: The 8086/8088 Family Architecture, Programming and Design	Yu-cheng Liu and Glenn A. Gibson	2nd	Prentice-Hall of India	2011
2.	Microprocessors and Microcomputer-Based System Design	Mohammad Rafiquzzaman	2nd	CRC Press	1995
3.	Advanced Microprocessor and Peripherals	K Bhurchandi , A. K. Ray	3rd	McGraw Hill Education	2017
4.	The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4 and Core2 with 64 - bit Extensions	Barry B. Brey	8th	Pearson	2012

Course code	**:**	CSE-358	**Credit**	**:**	1
Title	**:**	Microprocessors and Assembly Language Lab	**Prerequisite**	**:**	None
Type	**:**	*Laboratory Work*	**Contact hours**	**:**	26

Rationale

This lab course is designed to teach students about basic principles of microprocessor and assembly language programming skills and real time applications of Microprocessor as well as microcontroller.

Lab Objectives

- To introduce students with the internal architecture of the Microprocessor.
- To give a basic knowledge of programming in Assembly language
- To give a practical experience of how to program with the microprocessor
- To utilize the Intel 8086-80586 instruction set and will perform programming exercises in MS-DOS mode

Lab Outcome

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

- Write programs and solve problems in Assembly language
- Compare assembly programs with other programming languages

Exp. #	Title and Descriptions
1	Basics of Assembly Language Programming.
2	Introducing with Processor Status and Flag Register.
3	Programming with Flow Control Instructions.
4	Programming with Logic, Shift and Rotate Instructions.
5	Programming with Multiplication and Division Instructions.
6	Programming with Arrays and Addressing Modes.
7	Programming with String Instructions.
8	Introducing with Procedures.
9	Introducing with Files.

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
Core i5, 1.8 GHz, 4 gig RAM, 500 meg disk space	MASM / TASM / Assembly Emulator

---

Course code	**:**	CSE-359	**Credit**	**:**	3.0
Title	:	Computer Networks	**Prerequisite**	:	Data Communication
Type	:	*Theory*	**Contact hours**	:	39

Rationale

This course gives students an opportunity to learn about computer network organization and implementation, theoretical understanding of computer networks and finally gaining practical experience in designing communication protocols, installation, monitoring, and troubleshooting of current LAN systems.

Course Objectives

The objectives of the course are to

- Make students familiar with protocol stack of different network.
- Provides the way of channel sensing, channel sharing and transmission of IP and TCP packet.
- Give the idea of some real applications like DNS; Electronic Mail; The World Wide Web; Multimedia etc.
- Provide the technique of ensuring security of a network at application layer

Students Learning Outcomes

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

- Design and analyze LAN for an educational/research institute or office.
- Simulate LAN traffic at MAC layer
- Configure different applications like DNS, SMTP, HTTP servers etc.
- Apply security to networks of a corporate office

Course Description

#	Title and Descriptions
1	**Introduction** Basic concept of computer network; Network classification, structure/topology; Protocol Hierarchies; Open System Interconnection; Example networks.
2	**Physical layer** Guided physical transmission media, baseband and passband transmission, PSTN; circuit and packet switching, Mobile telephone system; ADSL.
3	**Data Link Layer** Data link layer design issues; Framing, Error detection and correction; Elementary data link protocols; Sliding window protocols; The data link layer in the Internet.
4	**Medium Access Sub-layer** Multiple Access Protocols: ALOHA; CSMA/CD Protocol; Collision-free protocols; Wireless LAN protocols; Ethernet; Bluetooth; Datalink layer switching.
5	**Network Layer** Network layer design issues; IP address and subnet masking, Routing algorithms; Congestion control algorithms; Internetworking; Network layer in the internet; Network layer in ATM networks.
6	**Transport Layer**

	The transport service; Elements of transport protocols; three-way handshake; TCP Congestion Control, The internet transport protocols; Application Layer: DNS-Domain Name System; Electronic Mail; The World Wide Web; Multimedia.
7	**Network Security** Secrecy; cryptography, substitution and transposition cipher; public and private key; RSA; basic of digital signature; message digest; Secure Hash Algorithm.

Recommended Books

1.	Computer Networks	Andrew S. Tanenbaum	5th	Prentice Hall	1988
2.	Data Communications and Networking	Behrouz A. Forouzan	5th	Mcgraw-Hill Forouzan Networking Series	2007
3.	Data and Computer Communications	William Stallings	8th	Pearson Prentice Hall	2007
4.	Internetworking with TCP/IP	Douglas E. Comer	5th	Prentice Hall of India	2013

Course code	**:**	CSE-360	**Credit**	**:**	1.0
Title	:	Computer Networks Laboratory	**Prerequisite**	:	Data Communication
Type	:	*Laboratory Work*	**Contact hours**	:	26

Rationale

This lab course is designed for the students to achieve a hands on experience on basic computer network. Theoretical lectures are completed by lab practice where theoretical knowledge is applied in an appropriate software environment. Students become familiar with the basic protocols of computer networks and can learn how these protocols can be used appropriately to assist in network design and implementation.

Lab Objectives

To configure different types of network using simulation or network hardware/software

Lab Outcome

Upon successful completion of this lab, the students should be able to:

- Configure practical network of an organization
- Have a basic knowledge of the use of cryptography and network security
- Identify deficiencies in existing protocols and formulate new and better protocols
- Understand the issues surrounding Mobile and Wireless Networks
- Have a working knowledge of datagram and internet socket programming

Exp. #	Title
1	Verification of Some Fundamental Commands of Network Connections
2	Packet through HUB, Bridge and Switch with basic client server model
3	Configuration of a Router from Command Line Interface (CLI)
4	Connection multiple routers and incorporation of host between two routers
5	VLAN Configuration with layer 3 switch and router
6	Implementation of SMTP
7	Verification of Telnet and Secure Shell
8	DNS Server and Web Browsing
9	Observation of FTP and HTTP protocols
10	IP telephony system
11	Implementation of OSPF (Open Shortest Path First) Algorithm
12	Implementation of RSA algorithm for text/image transmission

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
LAN card, cables connectors, switch, HUB, Router	CISCO Packet Tracer, Matlab 18

Course code	**:**	CSE-362	**Credit**	**:**	1
Title	:	Web Design and Programming Lab-II (JSP/Python)	**Prerequisite**	:	None
Type	:	*Laboratory Work*	**Contact hours**	:	26

Rationale

This lab course is designed to make students efficient in Web Design and Programming with Python through a hands-on guide to object-oriented python web programming, working with multiple types of servers, databases and web frameworks.

Lab Objectives

This course is an introduction to server side programming for the World Wide Web. We'll examine Server side technologies in depth:

- Server side programming with Python and Java
- Understanding popular framework for Python and Java for web application
- Programming in Python and JAVA with framework
- Understanding security issues and its protection.
- Structure Query Language (SQL) for interacting with databases

Lab Outcome
After completing this Laboratory course, the students will have: 1. Detail knowledge about server side programming 2. Securing web application 3. Practical experience of Python and JSP programming

Exp. #	**Title and Descriptions**
1	Understanding and Programming with Python-I
2	Understanding and Programming with Python-II
3	Understanding and Programming with Python-III
4	Installing Django framework
5	Programming with Django Admin
6	Django form and its registration
7	Development and programming with Django model
8	Understanding and Programming with JSP-I
9	Understanding and Programming with JSP-II
10	Understanding and Programming with JSP-III
11	Understanding Spring Framework
12	Framework application with form
13	Framework application with Database

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
High configuration PCs equipped with required software	pyCharm and Eclipse

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

Forth Year First Semester

Course code	**:**	CSE 400	**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

Course Description

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

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

Course code	**:**	CSE 401	**Credit**	**:**	3.0
Title	**:**	Theory of Computation and Compiler Design	**Prerequisite**	**:**	Discrete mathematics, Data structures, and Algorithms
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Understanding the inherent capabilities and limitations of computers is a fundamental question in computer science. To answer this question, we will define formal mathematical models of computation, and study their relationships with formal languages. Topics will consist of three central areas of the theory of computation: automata, computability, and complexity.

Course Objectives

The learning objectives of this course is to introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability.

Students Learning Outcomes

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

- Define languages by abstract, recursive definitions and by regular expressions.
- Design a finite automaton to recognize a given regular language.
- Transform a language into regular expression or finite automaton or transition graph.
- Define deterministic and nondeterministic finite automata.
- Prove properties of regular languages and classify them.

Course Description

#	Title and Descriptions
1	**Language theory** Languages in the abstract, Kleene closure, Recursive definition, Arithmetic expression-language, Languages associated with regular expression, Finite languages
2	**Finite automata** Defining languages by Finite Automata, Finite Automata and their languages, EVEN-EVEN languages
3	**Transition graph** Defining transition graphs, Generalized transition graphs, Nondeterminism, Turning transition graphs into regular expressions, Converting regular expressions into Finite Automata
4	**Regular languages** Closure properties, Complement and Intersection, Pumping Lemma
5	**Context free grammars** Define languages, Parse trees, The Total Language Tree of the CFG, Regular grammar, Ambiguity, Chomsky

	Normal Form, Derivations
6	**Context free languages** Self embeddedness, Pumping lemma, Closure properties
7	**Turing machines** Define Turing machine, Post Machines, Simulating PM on a TM, Simulating TM on PM, Phrase structure grammar, Church thesis

Recommended Books					
1.	Introduction to the Theory of Computation	Michael Sipser	2nd	McGraw-Hill	2012
2.	Introduction to Automata Theory, Languages, and Computation	John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman	3rd	Pearson Education	2006
3.	An Introduction to Formal Languages and Automata	Peter Linz	6th	Jones & Bartlett Publishers, Inc.	2016

Course code	**:**	CSE 403	**Credit**	**:**	3.0
Title	**:**	Software Engineering and Information System Design	**Prerequisite**	**:**	Programming language, Algorithms
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Software Engineering and information system design is difficult to teach and to learn in a University environment. Most of the important lessons of software engineering and information system design only apply to large scale software development, i.e. to systems that are bigger than it is possible for a small group of students to build in one or two terms. The skills needed to develop such systems have very little to do with the type of programming that most students learn at University. In fact, software engineering has very little to do with programming at all. Software Engineering is about the discipline needed to develop high quality software that can be understood, maintained and adapted over long periods of time by many different people. ‘Quality’ is a key word here. The notion of quality for software is different from the notion of quality for other types of engineered systems (electrical, mechanical, etc), because software is different from physical systems (during the course we'll examine why). The notion of quality for software is also different from the notion of quality for programs that students build on programming courses. An understanding of what software quality really means is central to understanding what software engineering is all about.

Course Objectives

- Knowledge of basic S/W engineering methods and practices, and their appropriate application.
- A general understanding of software process models and design patterns.
- Understanding of software testing approaches.
- Understanding of the role of project management including planning, scheduling, software risks, evolution management and good quality software.

Students Learning Outcomes

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

- Work collaboratively in a team environment to develop a software system to meet desired needs within realistic constraints, specific time frame and budget.
- Prepare technical documentations and make presentations on various aspects of a software development project, including the technical aspects (architecture, design, quality assurance) as well as the managerial aspects (planning, scheduling, and delivery).
- Use communication tools to effectively collaborate with team members in different countries with different cultural backgrounds.

Course Description

#	Title and Descriptions
1	**Software and Information System concepts** Concepts of Software Engineering, System Development life cycle, Different phases of Software System and Information Development, Different types of information, Legacy Software, Qualities of information, System Concepts and information system environment, Types of Systems, System dependability, Safety and Security, Systems engineering, Illustration of a dynamic Personnel Information System Model
2	**Software Process analysis** Process framework, The Capability Maturity Model Integration(CMMI), Process activities, Process patterns, Process assessment, Personal software process(PSP), Team software process(TSP), Product and process, Management activities, Project Management Concepts, Software process and project Metrics, Software Project Planning, Feasibility analysis.
3	**Development models** The waterfall model, Incremental model, RAD, Evolutionary process model: Prototyping, Spiral model, The concurrent development model, Aspect-oriented software development, Agility, Extreme programming, Adaptive software development(ASD), Dynamic systems development model(DSDM), Scrum, Crystal, Feature driven development(FDD), Agile modeling(AM), System Modeling: Hatley-Pirbhai Modeling, System Modeling with UML
4	**Requirement analysis and system modeling** Requirement engineering, Development of use cases, Analysis modeling, Data modeling, Object-Oriented Analysis, Scenario based modeling: Activity diagram, Swimline diagram, Flow-oriented modeling: DFD, Control flow model, Class based modeling, Behavioral model, Design concepts and principles, Design Quality, Architectural design, User Interface design, Object Oriented software development and design:

	Iterative Development and Unified Process.
5	**Object oriented analysis and different diagrams** UML diagrams, Interaction and Collaboration Diagram for designing Software. Designing Objects with responsibilities, Architectural design, Architecture trade-off analysis and complexity, Distributed Systems architecture, Elaboration using System Sequence Diagram, Application architecture: Data, transaction, event and language processing systems, Mapping data flow into architecture, Domain Model, Visualizing concept classes, Real time software design, User interface design
6	**Software design pattern** GRASP patterns with General Principles in assigning responsibilities: Information expert, Creator, Low Coupling and High Cohesion, Creating design class diagrams and mapping design to codes. Advanced GRASP patterns: Polymorphism, Pure Fabrication, Indirection, Project Variation, GoF Design Patterns: Adapter, Factory, Singleton, Strategy, Composite, Facade, and Observer.
7	**Software testing** Test case design, Testing Patterns, Software Testing: White Box and Black Box testing, Basis Path Testing, Component testing. Testing for specialized environment, Software testing strategies: Unit Testing, Integration Testing, Validation Testing, System Testing, Object oriented testing methods, Control Structure testing, Art of debugging, Test automation, Critical systems validation.

Recommended Books					
1.	Software Engineering: A practitioner's approach	Roger S. Pressman	7th	McGraw-Hill	2009
2.	Software Engineering	Ian Somerville	10th	Pearson	2015
3.	Systems Analysis and Design: An Object-Oriented Approach with UML	Alan Dennis, Barbara Haley Wixom, David Tegarden	5th	Wiley Publishing	2015

Course code	**:**	CSE 404	**Credit**	**:**	1
Title	**:**	Software Engineering and ISD Laboratory	**Prerequisite**	**:**	Database, Basic Programming
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale
The course attempts to foster an understanding of software development and quality: what it is, and how to achieve it. We do this through the use of a team project running throughout the course, in which teams trade software modules with one another. By attempting to understand, assess, and modify one another's programs, students will gain insight into the nature of software quality, and why an ability to program is not sufficient for the construction of high quality software.

Lab Objectives
• To be familiar with the state-of-the art knowledge of software engineering and UML in an interactive manner. • Present case studies to demonstrate practical applications of different concepts in order to solve small, real life problems.

Lab Outcome
After successful completion of this course, students should be able to: • Generate the requirements, and analysis and design models based on the analysis for their projects. • Participate in preparing the project plan. • Create and specify a software based on requirements, architecture and design.

Lab Course Description

Exp. #	**Title**
1	Identify Project scope, Objectives and Infrastructure of a particular project. Develop software requirements specification (SRS) document.
2	Develop DFD model (level-0, level-1 DFD and Data dictionary) and ER diagram of the project.
3	Perform the user's view analysis: Use case diagram.
4	Draw the structural view diagram: Class diagram, Object diagram.
5	Draw the behavioral view diagram: Sequence diagram, Collaboration diagram
6	Draw the behavioral view diagram: State-chart diagram and Activity diagrams.
7	Draw the Implementation Diagrams: Component diagram and Deployment Diagrams
8	Add interface to the class diagram and implement the design by coding.
9	Perform various techniques for testing using the testing tools.

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
High configuration Desktop computers	For UML diagrams: Rational Rose software, Umbrello, Coding: Netbeans IDE, Codeblocks, Java, C# Testing tool: Junit, Winrunner Project management tool: Wrike

---

Course code	**:**	CSE 405	**Credit**	**:**	3.0
Title	**:**	Digital Image Processing	**Prerequisite**	**:**	Basic Programming
Type	**:**	Theory	**Contact hours**	**:**	39
Rationale					

Visual information plays an important role in many aspects of our life. Much of this information is represented by digital images. Digital image processing is ubiquitous, with applications including television, tomography, photography, printing, robot perception, and remote sensing. This is an introductory course to the fundamentals of digital image processing. It emphasizes general principles of image processing, rather than specific applications.

Course Objectives

- To introduce students to modern digital image processing and its scope.
- To develop detailed and sound understanding and theoretical background on digital image processing.
- To learn the working principles of image acquisition modalities.
- To develop the skill to utilize the tools and methods used in digital image processing.
- To promote the programming skill to implement the basic algorithms/tasks in digital image processing.
- To grow the skill to deal with the real-world problem and to be able to link between the theory and practical realization.

Students Learning Outcomes

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

- Define and comprehend the basic principles of digital image processing
- Work with and apply commonly available tools and techniques efficiently solve an image processing problem.
- Explain the inner workings of general image acquisition modalities.
- To program fundamental image processing algorithms.

Course Description

#	Title and Descriptions
1	**Image, representation and Image formation** Introduction, images and pictures, image files, file types, resolution, color space, bit planes, image display, mathematics and engineering of image formation, human eye, cameras and other image processing modalities, etc.
2	**Introduction to MATLAB, Octave, Python etc.** *MATLAB/Octave*: Introduction, basic use, variables and workspace, matrices, plots, online help, programming. *Python*: Basic use, arrays, graphics and plots, programming.
3	**Point and Neighborhood Processing** Introduction, arithmetic operations, histograms, lookup tables, filtering, low and high pass filters, Gaussian filters, Region of interest (ROI) processing, etc.
4	**Image Transforms and Image Restoration** Two dimensional discrete Fourier transform (DFT), homomorphic filtering, frequency domain filtering, discrete wavelet transform, image denoising, filtering, and compression using wavelets, noise, cleaning salt and pepper noise, cleaning gaussian noise, removal of periodic noise, inverse filtering, Wiener filtering.
5	**Mathematical Morphology and Image Segmentation**

	Introduction, basic ideas, dilation & erosion, opening & closing, Introduction to image segmentation, thresholding, edge detection, second derivative, the Canny edge detector, corner detection, The Hough and Radon transform, etc.
6	**Color Processing, Image coding, compression and Special Effects** Color, color models, manipulating color images, pseudocoloring, processing of color images, lossless and lossy compression, Huffman coding, run length encoding, dictionary coding, the JPEG algorithm, polar coordinate, ripple effect, general distortion effects, pixel effects, effects on color images, etc.
7	**Image Features & Image object Classification** Landmarks and shape vectors, image features, principal component analysis, moments, texture, Gabor feature, dimensionality reduction, supervised and unsupervised classification, linear discriminant functions, eigenvalue and eigenvector, Bayesian classification, k-means clustering, support vector machine (SVM), etc.

Recommended Books

1.	A Computational Introduction to Digital Image Processing	Alasdair McAndrew	2nd	Chapman and Hall/CRC	2015
2.	Digital Image Processing	Rafael C. Gonzalez, Richard E .Woods	4th	Pearson	2017
3.	Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab	Chris Solomon, Toby Breckon	1st	Wiley	2011
4.	The Art of Image Processing with Java	Kenny A. Hunt	1st	CRC Press	2010

Course code	**:**	CSE 406	**Credit**	**:**	1
Title	**:**	Digital Image Processing Laboratory	**Prerequisite**	**:**	Basic Programming
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

The course will contribute to consolidation of students' engineering and mathematical knowledge and programming skills, extension of theoretical knowledge and practical skills to solve multidimensional visual signal processing problems, and the development of their ability to work on multidisciplinary issues in diverse areas of digital imaging and image processing applications as aforementioned.

Lab Objectives

- To train students to program using programming languages (or tools) MATLAB/Python
- To give practical experience in solving image processing related problems using computers.
- To develop ability to program basic algorithms in digital image processing.

Lab Outcome

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

- Know about the programming languages and tools to use in digital image processing.
- Solve real world digital image related problems.
- Implement/Program fundamental algorithms in digital image processing field.

Lab Course Description

Exp. #	Title
1	Programming using MASTLAB/Python etc.
2	Drawing Image Histogram and Performing Histogram Equalization
3	Filtering Images
4	Transforming Images
5	Segmenting Images
6	Restoring Images
7	Applying Morphological Operations on Images
8	Processing Color Image and Performing Special Effects
9	Extracting Image Features and Classify Image object

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
Camera/Webcam, GPU, FHD/4K Monitor, Multicore processor (CPU), etc.	MATLAB/Octave/Python

---

Course code	**:**	CSE 407	**Credit**	**:**	3.0
Title	**:**	Wireless Networks	**Prerequisite**	**:**	Data Communication
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

This course is intended to give students an outline of how wireless communication and computer networks work "above the physical layer". This includes the interoperability of wireless networks such as WiMax and WiFi to provide WiFi on trains etc. How wireless sensor networks gather and report physical parameters including body sensor networks. We also look at the evolution of cell phone networks from GSM->GPRS->3G->LTE->5G.

Course Objectives

- Make students familiar with WLAN, its protocol stack: specially at physical and datalink layer
- Provides the design procedures of radio and switching of WAN and MAN
- Integrate LAN and MAN to enhance traffic capacity of the aggregate network
- Provide the promising features of 4G and 5G networks compared to 3G

Students Learning Outcomes

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

- Design and analyze WLAN, home RF and WSN for an educational/research institute or office.
- Implement MANET or WSN in any emergency situation (lack of infrastructure)
- Make radio and traffic plan a WAN or MAN maintaining minimum interferences and reasonable QoS for a practical network
- Understand protocol and access technique 4G network then students will be able to analyze the built in functions of MATLAB-18 for LTE to simulate such network to measure the performance under fading channel.
- Learn features of 5G mobile communications

Course Description

#	Title and Descriptions
1	**Classification of wireless network** Home RF and Bluetooth, IEEE 802.11 family, protocol stack of IEEE 802.11, frame format of Wi-Fi, CSMA/CA of MAC sublayer, binary exponential backoff algorithm
2	**Mobile ad-hoc network (MANET)** Destination-Sequenced Distance Vector (DSDV) protocol, Cluster-Head Gateway Switch Routing Protocol, Wireless Sensor Network (WSN), Sensor Node Structure, LEACH and DEEP Clustering Protocol
3	**Wide area network** concept of cell and cell cluster, co-site, adjacent channel and co-channel interferences of WAN, handover and roaming, channel allocation scheme
4	**Advanced Mobile Communication** Vision of IMT 2000, principle of CDMA/WCDMA, architecture of 3G mobile (UMTS) communication, satellite based mobile communications
5	**WiMAX** Development of IEEE 802.16, adaptive modulation and channel coding of WiMAX, BW allocation algorithms, Wi-Fi and WiMAX integrated network, 802.16 Protocol Stack, a security sublayer, MAC common part sublayer and Service Specific Convergence Sub-layer, TDD and FDD operation
6	**4G Technology** Development of 4G long-term evolution (LTE), femtocell deployment , OFDMA-based physical layer access and MIMO of LTE, architecture of LTE, LTE frame structure and RB
7	**5G Wireless Systems** Cognitive radio network, cooperative spectrum sensing, objectives of 5G mobile communication,activities of

	METIS, 5G Challenges,Massive MIMO, D2D and M2M communications, Moving Networks and Ultra-dense Networks of 5G.

Recommended Books					
1.	An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications	Christopher Cox	1st	John Wiley & Sons, Ltd	2012
2.	Computer and Communication Networks	Nader F. Mir	2nd	Pearson Education	2014
3.	5G Wireless Systems: Simulation and Evaluation Techniques (Wireless Networks)	Yang Yang, Jing Xu, Guang Shi, Cheng-Xiang Wang	1st	Springer	2018
4.	Cognitive Radio Communications and Networks	Alexander M. Wyglinski, Maziar Nekovee, Y. Thomas Hou	1st	ELSEVIER Inc.	2009

Course code	**:**	CSE 410	**Credit**	**:**	2
Title	**:**	Mobile Application Development Laboratory	**Prerequisite**	**:**	C++/Java
Type	**:**	Laboratory Work	**Contact hours**	**:**	52

Rationale

Mobile computing devices have become ubiquitous in our communities. In this course, we focus on the creation of mobile solutions for various modern platforms, including major mobile operating systems. You will be able to take advantage of a wide variety of resources in building your application more rapidly and to be more sophisticated. You will be exposed to relevant tools and resources in the class so you can implement a reasonable variation of the chosen application in the class.

Lab Objectives

- Identify and describe the characteristics and constraints of programming mobile applications and analyze and evaluate different technologies and platforms in mobile application development.
- Explore and develop sophisticated and robust applications on mobile device.
- Design and develop computing system to extend and enhance the capability of mobile applications.

Lab Outcome

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

- Work with software/hardware tools to develop, test and debug mobile applications.
- Design and develop sophisticated and robust mobile applications in major mobile platforms
- Apply current software technologies, framework architecture and standards used in mobile application

development; and publish and maintain these applications in the marketplace.

Lab Course Description	
Exp. #	**Title**
1	Getting Started with Android; Anatomy of Android Application Development
2	Java Review-1
3	Java Review-2
4	Installation and Configuration of tools required for Android Application Development, Test & Creating Simple App, use AVD and Emulator
5	Android Activity Lifecycle
6	Android views and Layouts
7	Android Intents and Intent filters
8	Android Fragments
9	Android Data and Storage APIs; Sharing Data between applications using Content Providers
10	Android cursors, libraries, and navigation drawers
11	Android services
12	Introduction to important AP's-1 (Networking, Web, and Location Based APIs)
13	Introduction to important AP's-2 (Android Multimedia and Telephony APIs
14	Getting Started with XCode
15	User Interfaces in iOS
16	Introduction to Objective-C
17	iOS Application Development
18	Course Project

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
Desktop PC in Windows/Linux/Mac OS with At least 4 GB RAM (8 GB Recommended)	Java, Android Studio, XCode

---

Course code	**:**	CSE 440	**Credit**	**:**	**2.0**
Title	:	**Research Project**	**Prerequisite**	:	None
Type	:	*Research Project*	**Contact hours**	:	-

Rationale
The course concentrates on creating links between theory and practice. It covers a wide variety of software and hardware technologies and their applications.

Course Objectives

- To develop knowledge of research methodologies in Computer science.
- To demonstrate an in-depth ability to approach problems in Computer science in a scientific manner.

Students Learning Outcomes

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

- Demonstrate knowledge of qualitative and quantitative research methods
- Demonstrate knowledge on how research projects are carried out
- Demonstrate knowledge on ethical considerations in research projects
- Demonstrate ability to formulate a scientific problem
- Demonstrate ability to find and evaluate relevant information for a scientific problem
- Demonstrate ability to analyze results with appropriate statistical methods
- Demonstrate ability to present results in a scientific manner

Course Description

#	Title and Descriptions
	The course is based on an individual research work including literature studies according to the study plan. An individual study plan will be commonly written by the supervisor and the student which serves as a project description. At the end of the practical work, the students will write a research report. A poster based on the research results will be designed, presented and discussed.

References

The reading materials are provided by the supervisor.

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

Forth Year Second Semester

Course code	**:**	CSE 450	**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

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

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

Course code	**:**	CSE 451	**Credit**	**:**	3.0
Title	**:**	Data Mining and Big Data Analysis	**Prerequisite**	**:**	Database
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Data Mining and Big Data Analytic studies algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data. It is currently regarded as the key element of a more general process called Knowledge Discovery that deals with extracting useful knowledge from raw data. The knowledge discovery process includes data selection, cleaning, coding, using different statistical and machine learning techniques, and visualization of the generated structures. The course will cover all these issues and will illustrate the whole process by examples. Special emphasis will be given to the Machine Learning methods as they provide the real knowledge discovery tools.

Course Objectives

- The fundamentals of Data mining techniques
- Analyzing a real world system, analyzing its data and information processing for decision making
- Details of knowledge discovery in databases (KDD) process, different data mining methods, e.g., classification, association analysis, clustering etc.
- Details of Data warehouse systems, models, architecture, OLAP operations and applications.
- Details of the basic concept of Big Data, Big data analytics, Digital Data, Data Store, Open Access Tools, and MapReduce programming.

Students Learning Outcomes

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

- Detail knowledge of data mining, data warehouse and data analytics fundamentals
- Detail theoretical knowledge of applications, algorithms, architectures and operations for data mining.
- Detail knowledge of various data mining methods in knowledge extraction from databases for decision making.
- Basic knowledge about the concepts of Big data analytics, Data Store, Using Open Access Tools, and MapReduce programming.

Course Description

#	Title and Descriptions
1	**Basic concepts of data mining and applications** KDD Process, Data Mining System Architecture, Data Mining Query Language, OLAM Architecture. Statistics and machine learning. Acquiring, parsing, filtering, mining, representing, refining and interacting with data. Genomic Microarray Data Analysis.
2	**Data Warehouses**

	Basic Concepts, Dimensions and fact tables, OLTP and OLAP, Data Preprocessing, architecture and operations, Conceptual Modeling, Usage
3	**Frequent pattern algorithms** Association Rule Mining, Sequential Pattern Mining, Mining frequent structures.
4	**Classification and Clustering Algorithms** Decision Tree Classification, Naive Bayesian Classification; K-means clustering. Data visualization. Scalable Data Mining algorithms and systems support, Parallel algorithms. Database integration, Data Locality Issues.
5	**Big Data Analytics Overview** Basic concept of Big Data, Big data analytics; Digital Data, Data Store;
6	**Open Access Tools** Introduction to Hadoop/MongoDB environment and its use in Big Data Processing; Database processing; Query processing.
7	**MapReduce programming** Introduction to MapReduce programming; Partitioner.

Recommended Books					
1.	Data Mining: Concepts and Techniques	Jiawei Han, Micheline Kamber	2nd	Elsevier	2010
2.	Introduction to Data Mining	P N Tan, M Steinbach, V Kumar	2nd	Pearson	2018
3.	Big Data and Analytics	Seema Acharya, Subhashini Chellappan	1st	Wiley	2015

Course code	**:**	CSE 452	**Credit**	**:**	1.0
Title	**:**	Data Mining and Big Data Analysis Laboratory	**Prerequisite**	**:**	Database
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

The goal of the course is to introduce students to the current theories, practices, tools and techniques in data mining. Because many topics and concepts in data mining are learned most efficiently through hands-on work with data sets, we will spend time with software analyzing and mining data. The goal is to gain a better understanding of how data mining is applied and what is involved in data mining projects.

Course Objectives

- How to prepare a Lab. Report.
- How to analyze a real world system, analyzing its data and information processing and knowledge extraction for decision making.
- Experimenting on data mining methods e.g., classification, association analysis, prediction, and clustering

using suitable algorithms with available data mining tools.

- Experimenting on the basic concept of Big Data processing using available Big Data Analytics tools.
- Experimenting on the MapReduce programming using available Big Data Analytics tools.

Students Learning Outcomes

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

- Obtain the detail knowledge of sample data preparation and knowledge extraction from the data using data mining tool based on their own project.
- obtain the detail experimental knowledge of applications of data mining algorithms using data mining tools on sample data.
- Learn experimental knowledge on Big Data processing using available Big Data Analytics tools
- Learn experimental knowledge on MapReduce programming using available Big Data Analytics tools.

Course Description

Exp. #	Title
1	Prepare a System Description using data oriented analysis for data mining applications of your project.
2	Experiment on Data characterization.
3	Experiment on Classification by applying classification algorithm using a Data mining tool, e.g. Weka.
4	Experiment on Association Rule Generation by applying a suitable algorithm using a Data mining too, e.g. Weka.
5	Experiment on Prediction using a Data mining tool
6	Experiment on Clustering using a Data mining tool, e.g. Weka.
7	Write program code in Java/C# for implementing any classification algorithm connecting a RDBMS database.
8	Experiment on Hadoop file handling commands
9	Experiment on Hadoop/MongoDB to implement a partitioner

Hardware and Software Requirements

H/W Requirements	*S/W Requirements*
High configuration PCs equipped with required software	Weka, Neuralware, Hadoop, MongoDB, Java, C#, MySQL, MS Excel etc.

Course code	**:**	CSE 453	**Credit**	**:**	3.0
Title	**:**	Artificial Intelligence	**Prerequisite**	**:**	Discrete Mathematics / Basic Programming
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Artificial Intelligence (AI) aims to make computers and information systems more "intelligent" to solve complex problems and provide more natural and effective services to human beings. AI has been a source of innovative ideas and techniques in computer science, and has been widely applied to many information systems. This course provides a comprehensive, graduate-level introduction to artificial intelligence, emphasizing advanced topics such as advanced search, reasoning and decision-making under uncertainty, and machine learning.

Course Objectives

- To describe the fundamentals of AI, logic, knowledge representation, organization, manipulation, inferencing, resolution, natural language processing and a general understanding of AI principles and practice.
- To understand how to build simple knowledge-based expert systems and various AI search optimization strategies (uninformed, informed, genetic algorithms)
- To expose the students to the AI programming tools and techniques for real-life problem solving.
- To design different types of AI agents and fuzzy-based systems

Students Learning Outcomes

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

- Will develop skills on modeling and analysis for duplication of human intelligence into machine. (e.g., search, logic, probability, reasoning) in solving real-life problems
- Will gain skills on designing different types of AI agents, optimization strategies and fuzzy-based systems
- Will help in achieving communication and presentation skills
- Will develop attitude to group dynamics and team work

Course Description

#	Title
1	**Introduction to Artificial Intelligence (AI)** History, success and failures, mission and vision, Intelligent Agents
2	**The Foundations of Logic, Formalized Symbolic Logic** First Order Predicate Logic (FOPL), Modus Ponen, Modus Tollen
3	**Knowledge Representation** General Concepts, Design and Representation: Frame, Semantic Nets
4	**Problem Solving by Searching and Stochastic Searching Tool** Uninformed and informed searches and Gaming Theory, Genetic Algorithms (GA)
5	**Introduction to Fuzzy Logic and fuzzy systems** Fuzzy and Crisp logic, membership functions, Fuzzy sets, hedges, Operations, Rules of inference, Defuzzification
6	**Expert Systems**

	Concepts, Need and justification for expert systems, Typical expert system architecture, Knowledge acquisition, Case studies: Rule based expert system, Neural expert systems, Neuro-fuzzy expert systems, Adaptive Neuro-fuzzy inference systems
7	**Natural Language Processing** Concepts, Components, Context-free grammer, Parsing, RTN

Recommended Books					
1.	Artificial Intelligence – A Modern Approach	Stuart J. Russel and Peter Norvig	2nd	Prentice-Hall	2003
2.	Introduction to Artificial Intelligence and Expert Systems	Dan W. Patterson	2nd	Prentice-Hall	2003
3.	Artificial Intelligence: A Guide to Intelligent Systems	Michael Negnevitsky	3rd	Pearson UK	2011

Course code	**:**	CSE 454	**Credit**	**:**	1.0
Title	**:**	Artificial Intelligence Laboratory	**Prerequisite**	**:**	Basic Programming
Type	**:**	Laboratory Work	**Contact hours**	**:**	26

Rationale

Artificial intelligence (AI) techniques are now being applied in many branches of engineering to solve problems and to provide intelligent interfaces for systems and equipment. This course provides an introduction to the four major AI techniques of rule-based expert systems, neural networks, genetic algorithms and fuzzy logic.

Course Objectives

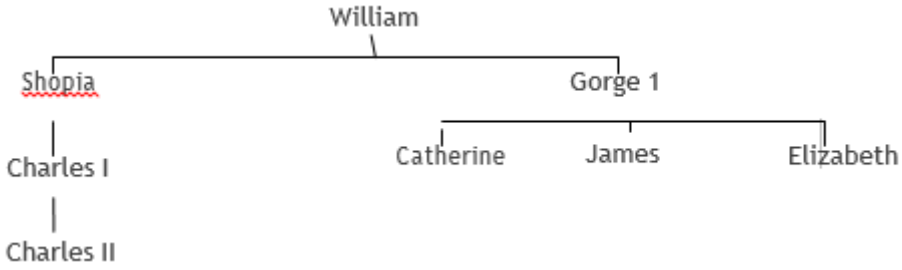
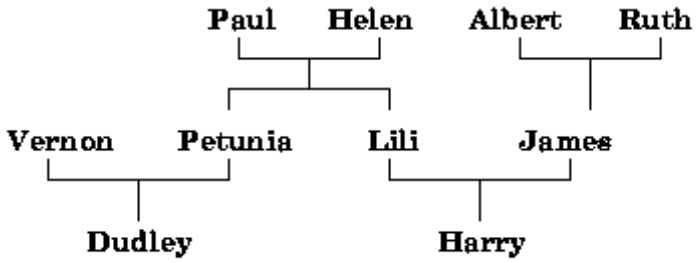
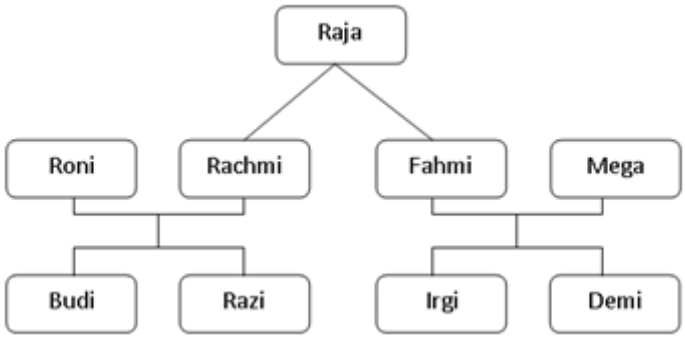
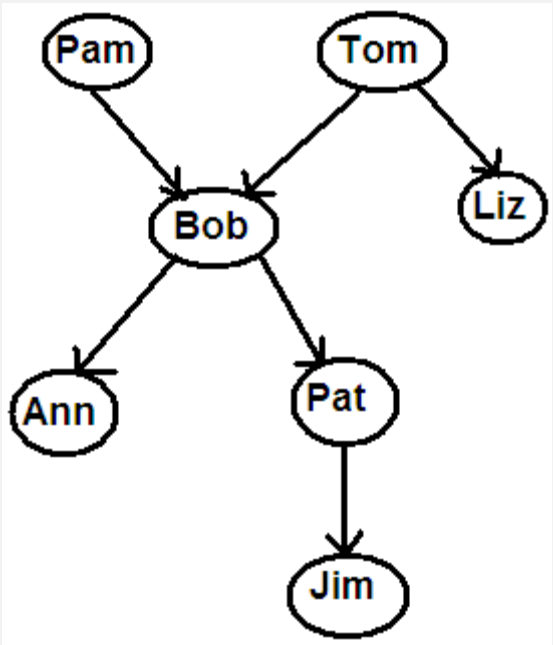
- To expose the students to the AI programming tools and techniques for real-life problem solving
- To design different types of AI agents, search and optimization strategies and knowledge- based systems

Students Learning Outcomes

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

- Develop skills in designing different types of search and optimization strategies and knowledge-based systems
- Gain skills on AI programming tools
- Achieve communication and presentation skills
- Develop attitude to group dynamics and team work

Course Description

Exp. #	Title
1	 Add rules for brother, sister, father, mother, grandparent, aunt, uncle.  Add rules for brother, sister, father, mother, grandparent, husband, wife.  Add rules for brother, sister, father, mother, grandparent, husband, wife.
2.	This following family tree is given  Do the following: i) Show all of the parents of each child. ii) Show father of a child.

	iii) Show grandfather of a grandchild.
3	Write a prolog program to find the sum of all numbers in a list. Write a prolog program to find the length of a list. Write a prolog program to find the maximum number of a list.
4	Consider the following database Codd lectures in course cse9020 and cse9314 Backus lectures in course cse9021 Rictchie lectures in course cse9201 Minsky lectures in course cse9414 Backus lectures in course cse9311 Fred studies in course cse9020 Jack studies in course cse9311 Jill studies in course cse9314 Jill studies in course cse9414 Henry studies in course cse9414 Henry studies in course cse9314 Course cse9020 is offered in semester summer. Course cse9201 is offered in semester fall. Course cse9021 is offered in semester summer. Course cse9414 is offered in semester spring. Course cse9311 is offered in semester spring. Course cse9314 is offered in semester fall. Do the following: i) Write a rule to show the course lists of any teachers. ii) Write a rule to find out who is taught by a teacher. iii) Write a rule to report on Fred's summer courses. iv) Write a rule to show if a teacher taught in summer or not? v) Show each students course name.
5	Basic Arithmetical Operations of Prolog. Write a rule to show the - i) summation of two numbers. ii) subtraction of two numbers. iii) multiplication of two numbers. iv) division of two numbers. v) maximum number from given three numbers. vi) X is raised to Y power. vii) remainder of two numbers. viii) bitwise AND operation between two numbers. ix) bitwise OR operation between two numbers. x) bitwise XOR operation between two numbers. xi) bitwise left shift operation of the number. xii) bitwise right shift operation of the number. xiii) bitwise complement operation of the number.
6	Advanced Arithmetical Operations of Prolog. Write a rule to show the - i) Roots of a quadratic equation.

	ii) GCD (Greatest Common Divisor). iii) Area of a triangle. iv) Area of a circle Finding Factorial Write a rule to show the – i) Factorial of a given integer. Find Total Cost from List The following two lists are given i) Product name ii) Product cost Write a rule to show the – Cost of a given product
7	Write a program to find the path of a desired node using BFS.
8	Write a program to find the path of a desired node using DFS.
9	Write a program to solve travelling sales man problem using genetic algorithm.

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
High configuration PCs equipped with required software	SWI-Prolog

---

Course code	**:**	CSE 455	**Credit**	**:**	3.0
Title	**:**	Software Quality Assurance	**Prerequisite**	**:**	Software Engineering
Type	**:**	Theory	**Contact hours**	**:**	39

Rationale

Building on previous exposure to the fundamentals of the software process, this course focuses on techniques for ensuring software quality. Here, quality assurance is viewed as an activity that runs through the entire development process: understanding the needs of clients and users, analyzing and documenting requirements, verifying and validating solutions through testing.

Course Objectives

- Understand how to detect, classify, prevent and remove defects
- Understand how to conduct formal inspections, record and evaluate results of inspections
- Know how to choose which metrics to collect and use them to make predictions
- Choose appropriate testing strategies and develop test cases
- Be able to use Z to formally specify a system and write proofs for algorithms.

Students Learning Outcomes
After successful completion of this course, students should be able to: • critically evaluate alternative standards, models and techniques aimed at achieving quality assurance in a variety of software development environments • propose and defend innovative solutions to software quality assurance and measurement problems in the context of various software development environments • critically evaluate leading edge approaches in software development and attendant quality assurance methodologies, presenting the research using Harvard referencing

Course Description	
#	**Title**
1	**Software Quality Challenges and Factors** Uniqueness of SQA, Environments, Classification of software errors, Software quality, Requirements, Classifications, Product operation, revision and transition, Active models of software quality factors
2	**Components of SQA and Contract review** Architecture, Pre-project components, Project life cycle, Infrastructure, Management SQA components, SQA standards, system certifications, and assessment components, Organizing for SQA Contract review: Introduction, Process and its stages, Objectives, Implementation, Subjects, Contract review for internal projects
3	**Development and Quality plans, Integrating quality activities and Review** Objectives, Elements, Development plan and quality plan for small projects and for internal projects, Software development methodologies, Factors affection intensity of QA activities, Verification, Validation and qualification, Review: Objectives, Formal design, Peer reviews, Comparison, Expert opinions
4	**Software testing strategies, implementation and maintenance components** Definition and objectives, Strategies, Classifications, White box testing, Black box testing, Testing process, Test case design, Automated testing, Alpha and beta site testing programs, Foundation of high quality, Pre-maintenance components, Maintenance SQA tools, CASE tools, Contribution of CASE tools to software product and maintenance quality and to improve project management
5	**Software Quality infrastructure components** Procedures and work instructions, Supporting quality devices, Staff training and certification, Creative and preventive actions, Configuration management, Documentation control
6	**Management components of software quality** Project progress control: Components, Progress control for internal projects and external participants, Implementation, Computerized tools Cost of software quality: Objectives, Models, Applications, Problems
7	**Standard, certification and assessment** Quality management standards: Scope, ISO 9001 and ISO 9000-3, Certification, CMM and CMMI, Bootstrap methodology Project process standards: Structure and content, IEEE/EIA Std 12207-Software life cycle process, IEEE Std

	1012-verification and validation, IEEE Std 1028-reviews

Recommended Books					
1.	Software Quality Assurance: From Theory to Implementation	Daniel Galin		Pearson	2003
2.	Software Quality Assurance	R. Chopra	Cdr	Mercury Learning	2018
3.	Software Quality Assurance	Claude Y. Laporte, Alain April	11th	Wiley-IEEE Computer Society Pr	2013
4	Software Quality Assurance: Principles and Practices	Nina S Godbole		Alpha Science International	2008

Course code	**:**	CSE 457	**Credit**	**:**	3.0
Title	**:**	**Machine Learning**	**Prerequisite**	**:**	**Probability and Statistics/ Discrete Mathematics/Basic Programming**
Type	:	Theory	**Contact hours**	:	39

Course Objectives

- To expose the students to the basic machine learning tools and techniques;
- To understand, implement and apply the machine learning techniques on solving real problems, designing a novel algorithm for supervised or unsupervised learning;
- To extend, improve, or speeding-up some existing machine learning algorithms.

Students Learning Outcomes

- Will develop skills in applying machine learning techniques on solving real problems;
- Will help in achieving communication and presentation skills;
- Will develop attitude to group dynamics and team work.

Course Description

Topics	Descriptions
1	Regression, Discriminative Algorithms
2	Bayesian classifier, Decision Tree Learning
3	Neural Networks and Deep Learning
4	Support Vector Machine
5	K-means Clustering, EM and SOM
6	PCA Learning

7	Singular Value Decomposition (SVD) and Reinforcement Learning

Recommended Books

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006, (ISBN-13: 978-0387-31073-2).
2. Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014, (ISBN 978-1-107-05713-5).
3. Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press, 2009, (ISBN 9781466583283)

Course code	**:**	CSE-458	**Credit**	**:**	1.0
Title	:	**Machine Learning Laboratory**	**Prerequisite**	:	Probability and Statistics/ Discrete Mathematics/Basic Programming
Type	:	LAB Works	**Contact hours**	:	26

Course Objectives

- To expose the students to the basic machine learning tools and techniques;
- To understand, implement and apply the machine learning techniques on solving real problems, designing a novel algorithm for supervised or unsupervised learning;
- To extend, improve, or speeding-up some existing machine learning algorithms.

Students Learning Outcomes

- Will develop skills in applying machine learning techniques on solving real problems;
- Will help in achieving communication and presentation skills;
- Will develop attitude to group dynamics and team work.

Experiments: Experiments will be set on Machine Learning Theory Course

1 Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

2 Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart-Disease Data Set.

3 Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm and also SOM. Compare the results of these two algorithms and comment on the quality of clustering.

4 Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.

5 Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

6 Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

7 Build an Artificial Neural Network by implementing the CNN algorithm and test the same using appropriate data sets.

8 Build an Artificial Neural Network by implementing the deep CNN algorithm and test the same using appropriate data sets.

Course code	**:**	CSE 460	**Credit**	**:**	2
Title	**:**	**IoT Laboratory**	**Prerequisite**	**:**	
Type	**:**	Laboratory Work	**Contact hours**	**:**	52

Rationale

The Internet of Things (IoT) is a popular buzzword right now, but unlike many fads which have come and gone, the Internet of Things describes an important trend which is having lasting effects on society at large. The term itself, “Internet of Things”, is used to mean a variety of ideas, depending on the motivation and background of the speaker. This course will start by providing a definition of the term. We will talk about how various trends have enabled the Internet of Things, and how it changes the way design is performed. We will also discuss some of the ramifications that IoT is having on society today.

Lab Objectives

- To enable the interconnection and integration of the physical world and the cyber space
- To give core concepts of IoT, role and scope of smart sensors for insuring convergence of Technologies and multidisciplinary engineering practices, Machine Intelligence Quotient.
- To gain the idea of Big data predictive analytics and transformation from IT to IOT.
- Awareness of IoT related cyber legislation.
- To gain a hands-on experience on IoT based project

Lab Outcome

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

- Relate the interconnection and integration of the physical world and the cyber space
- Have a core concepts of IoT, its role and scope of smart sensors for insuring convergence of Technologies and multidisciplinary engineering practices, Machine Intelligence Quotient.
- Develop a problem oriented project on IoT.

Lab Course Description	
Exp. #	**Title**
1	Introduction to IoT and IoT equipment and turn an LED ON and OFF with Arduino
2	Read analog and digital device and print its state out to the Arduino Serial Monitor. Demonstration of the use of analog output to fade an LED. Reading an analog input and prints the voltage to the Serial Monitor.
3	Analog and digital communication with Arduino
4	Reading and writing on SD cards or EPROM with Arduino
5	Demonstration of keyboard and joystick interface with Auduino
6	Demonstration of IR and sonar sensor with Auduino
7	Use of Bridge Library like: Datalogger, HTTP client, mailbox read SMS wit Arduino
8	Demonstration of Motor Shield with Auduino
9	Demonstration of Steeper motor Shield with Auduino
10	Demonstration of Servo motor Control Shield with Auduino
11	Demonstration of Bluetooth Shield with Auduino
12	Demonstration of Ethernet Shield with Auduino
13	Demonstration of Wifi Shield with Auduino
14	Demonstration of GSM/GPRS Shield with Auduino
15	Control of Arduino Robot
16	Experiment with raspberry-pi: [Example experiment]
17	Experiment with raspberry-pi: [Example experiment]
18	Experiment with raspberry-pi: [Example experiment]

Hardware and Software Requirements	
H/W Requirements	*S/W Requirements*
Computer, Sensors, Arduino board, raspberry-Pi, Motors	Arduino 1.8.3 or Later

---

Course code	**:**	CSE 480	**Credit**	**:**	**3.0**
Title	:	**Research Project**	**Prerequisite**	:	None
Type	:	*Research Project*	**Contact hours**	:	**-**

Rationale
The course concentrates on creating links between theory and practice. It covers a wide variety of software and hardware technologies and their applications.

Course Objectives

- To develop knowledge of research methodologies in Computer science.
- To demonstrate an in-depth ability to approach problems in Computer science in a scientific manner.

Students Learning Outcomes

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

- Demonstrate knowledge of qualitative and quantitative research methods
- Demonstrate knowledge on how research projects are carried out
- Demonstrate knowledge on ethical considerations in research projects
- Demonstrate ability to formulate a scientific problem
- Demonstrate ability to find and evaluate relevant information for a scientific problem
- Demonstrate ability to analyze results with appropriate statistical methods
- Demonstrate ability to present results in a scientific manner

Course Description

#	Title and Descriptions
	The course is based on an individual research work including literature studies according to the study plan. An individual study plan will be commonly written by the supervisor and the student which serves as a project description. At the end of the practical work, the students will write a research report. A poster based on the research results will be designed, presented and discussed.

References

The reading materials are provided by the supervisor.