INSTITUTE OF INFORMATION TECHNOLOGY

UNIVERSITY OF DHAKA

http://www.iit.du.ac.bd/

CURRICULUM OF

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING (BSSE)

Effective from Session 2022-23

Copy sent to Academic Council.

Shabir
11/04/23

TABLE OF CONTENTS

PART A: INTRODUCTION

Vision Statement of Institute of Information Technology

To provide state-of-the-art education in the field of Information Technology and prepare future leaders who can create an impact in human lives through ethical and innovative use of technology.

Mission of Institute of Information Technology

Institute of Information Technology (IIT) aims to achieve the following missions:

M1	To provide world-class education and knowledge in Information Technology creating highly competent professionals who can solve real-world problems to meet national and global demands
M2	To contribute towards the creation of knowledge in Information Technology and related fields through research and innovation
M3	To collaborate with academic institutions, industry, and policy-makers for conducting research and building new technology
M4	To foster a strong sense of community, mutual respect and shared mission among students, faculty, staff, and alumni
M5	To instill in students strong moral and ethical principles that will enable them to contribute to the creation of a peaceful and prosperous society

Introduction to Institute of Information Technology

Institute of Information Technology (IIT), University of Dhaka started its journey in June 2001 to create efficient manpower in information technology. IIT currently offers Bachelor of Science in Software Engineering (BSSE), Master of Science in Software Engineering (MSSE), and Executive Master in Information Technology (MIT). Apart from its regular activities, it also produces state-of-the-art research and those are published in world-class journals and conferences.

IIT has a silent ambiance and is adjacent to the Dhaka University Science Library. The four-storey red ceramic IIT building contains seven software laboratories, one seminar library, and two large classrooms. The laboratories and classrooms are air-conditioned and equipped with multimedia, document camera systems, and Internet facilities.

Under the IT portfolio, skills in areas like data science, security, and especially software engineering are in demand in the job market. To fulfill the expanding software industry demands, IIT has come up with the undergraduate program titled "Bachelor of Science in Software

Engineering". Currently, IIT is affiliated with over 20 top software firms in Bangladesh and the course curriculum of IIT is a joint outcome of the academicians as well as the industry leaders.

Introduction to the BSSE Program

Title of the Program: Bachelor of Science in Software Engineering (BSSE)

Description of the Program:

Software engineering is a systematic and disciplined approach to develop software. It applies computer science, mathematics, general education and business principles and practices to the creation, operation, and maintenance of software systems. The Bachelor of Science in Software Engineering (BSSE) degree aims at the development and management of quality-measured software systems. It covers a detailed study that incorporates analysis and design, coding, testing, and deployment. The BSSE degree provides a systematic, work-integrated study aiming at software development, maintenance and operation. The degree will be particularly suitable for students with high technical aspirations and strong communication skills.

The BSSE program is a four-year industry-oriented degree. After studying for the first five semesters, the students are placed in the industry as interns in their sixth semester. This develops strong communication skills together with an outward worldly focus, positive personality, and business attitudes among the students. Upon return for their final year, students integrate their industry experience with the rest of their studies, which includes a year-long capstone project. The final year also gears the students towards attaining in-depth knowledge in one of several tracks related to software engineering, such as data science, information security, computer vision and advanced software engineering. This enables the students to combine their broad knowledge in software engineering with a focused understanding of a field of their interest.

Program Educational Objectives (PEOs) of BSSE

The graduates of our program will:

PEO1	Comprehend the basic knowledge, theories, principles, processes and procedures of a wide range of programming, information technology, and computational concepts.
PEO2	Have a positive mindset, interpersonal, leadership, and team-building skills to face global Software Engineering challenges while demonstrating integrity and ethical standards.
PEO3	Develop excellent analytical and critical thinking abilities so that they can put their knowledge into practice or pursue higher education.

PEO to Mission Statement Mapping

MISSION STATEMENTS	PEO1	PEO2	PEO3
M1	X		X
M2	X		X
M3		X	
M4		X	
M5		X	X

Program Learning Outcomes (PLOs)

PLO1	Apply knowledge of mathematics, computation, and software engineering fundamentals to solve complex software engineering problems
PLO2	Recognize, define, investigate existing literature, and examine intricate software engineering problems by utilizing fundamental concepts of mathematics, computation, and engineering sciences to arrive at well-supported conclusions.
PLO3	Create solutions for complex software engineering problems and develop systems, components, or procedures that satisfy specified requirements with suitable attention to societal, cultural, and environmental factors.
PLO4	Investigate complex problems and draw valid conclusions by utilizing research-based knowledge and methods such as designing experiments, analyzing and interpreting data, and synthesizing information.
PLO5	Apply modern software engineering and IT tools, prediction and modeling techniques, and appropriate resources to complex software engineering activities, while also recognizing their limitations.
PLO6	Utilize contextual knowledge to reason and evaluate societal, cultural, legal, and safety issues, and corresponding responsibilities relevant to professional engineering practices.
PLO7	Demonstrate an understanding of how professional software engineering solutions affect society and the environment, and exhibit knowledge of the importance of sustainable development.
PLO8	Adhere to the ethical principles and uphold the professional responsibilities and norms of software engineering practice.
PLO9	Work both independently and collaboratively as a member or leader of diverse teams in multidisciplinary settings.

PLO10	Communicate complex software engineering activities effectively with both the software engineering community and society as a whole. This involves the ability to create and interpret effective documentation, communicate effectively, and provide and receive precise instructions.
PLO11	Exhibit comprehension and familiarity with software engineering and management principles and implement them in the workplace.
PLO12	Acknowledge the necessity for and possess the readiness and capability to pursue independent and life-long learning in the widest scope of technological advancements.

Mapping PLOs with the PEOs

PEO/ PLO	PEO1	PEO2	PEO3
PLO1	X		X
PLO2	X		
PLO3	X		X
PLO4	X		
PLO5	X		
PLO6	X		
PLO7	X		X
PLO8			X
PLO9		X	
PLO10		X	
PLO11		X	
PLO12		X	X

PART B: BSSE CURRICULUM & COURSE DETAILS

The Bachelor of Science in Software Engineering (BSSE) Honours Program

(a) "Bachelor of Science in Software Engineering" (BSSE) is an undergraduate program under the Institute of Information Technology (IIT), University of Dhaka.

(b) The degree of Bachelor of Science in Software Engineering is conferred as a degree with Honours.

(c) BSSE degree is a terminal degree.

Admission in the Program

(a) Students will be admitted to the institute as per university rules.

(b) Each year application requirements will be defined by the central admission committee of the University of Dhaka.

(c) Institute can set specific requirements on subject-wise scores in admission test and/or HSC level.

(d) After paying admission, registration, yearly tuition, and other fees, a selected student has to complete his/her first admission and registration processes according to the university practices.

Duration of the Program

(a) BSSE is a 4 (four) academic years program. The program comprises 8 (eight) semesters, each having a duration of six academic calendar months.

(b) The maximum period of six consecutive academic years starting from the date of first admission in 1st year 1st semester.

Breakdown of Academic Years

Each academic year comprises 2 (two) semesters and 8 (eight) semesters are distributed as follows:

(a) 1st year comprises of 1st and 2nd semesters,

(b) 2nd year comprises of 3rd and 4th semesters,

(c) 3rd year comprises of 5th and 6th semesters, and finally

(d) 4th year comprises of 7th and 8th semesters.

Breakdown of a Semester

Each semester has a duration of 6 (six) months to be distributed as follows:

(a) Classes	: Fifteen weeks
(b) Preparatory Leave	: Maximum two weeks
(c) Semester Final Exam	: Three weeks
(d) Vacation	: Only the usual university's vacation will be applicable
(e) Result publications	: Within 60 (sixty) days (from the last theory exam date)

Studentship

(a) As long as a student is studying in this program he/she cannot get admission/study in any other department/institute of the university or any other educational institution. He/she may not even be able to serve in a public/private organization. If such an incident becomes apparent then the admission/degree of the student in question will be canceled.

(b) As long as a student gets the BSSE degree he/she has to get admitted or readmitted to the university by paying yearly tuition fees and other dues according to university rules as:

(i) admitted to next year if he/she got a promotion to next year

(ii) readmitted (maximum twice in the six consecutive academic years) to same year if he/she failed to get promotion to the next year.

Otherwise, his/her studentship will be canceled.

Requirements for the Award of the BSSE Degree

(a) The student must earn a minimum 147 credits including an internship program in a maximum of six consecutive academic years including the year of first admission into the program.

(b) The student must obtain CGPA of at least 2.5 to achieve the BSSE degree without an "F" grade in any course.

Definition of Credit

(a) For theoretical courses fifteen class-hour of fifty minutes each is defined as one credit.

(b) For practical or lab courses thirty lab hours of work is defined as one credit.

Course Load for BSSE Students

BSSE is a full-time course of study. Any student failing to take all credits (Please see *Section Semester-Wise Course Details*) in a regular semester will stand withdrawn from the program for that particular semester. An exception to this rule may be made only by the academic committee of the institute.

Credit Distributions for the 4-Year BSSE (Hons.) Degree

(a) Credits for core courses (Theory + Lab)	: 93 credits
(b) Credits for software project lab	: 6 credits
(c) Credits for general education	: 12 credits
(d) Credits for math & statistics	: 15 credits
(e) Credits for project	: 6 credits
(f) Credit for Internship	: 15 credits
Total Credits	: 147 credits

Grades and Grade Points

The current UGC-approved grading system applies as per university rules. Grades and grade points will be awarded on the basis of marks obtained in the written, oral and practical Exams according to the following scheme.

Marks	Letter Grade	Grade Point
80% and Above	A+	4.00
75% to < 80%	A	3.75
70% to < 75%	A-	3.50
65% to < 70%	B+	3.25
60% to < 65%	B	3.00
55% to < 60%	B-	2.75
50% to < 55%	C+	2.50
45% to < 50%	C	2.25
40% to < 45%	D	2.00
Less Than 40%	F	0.00
	I	Incomplete
	W	Withdrawn

(a) Only "D" or higher grade will be counted as credits earned by a student.

(b) A student obtaining an "F" grade in any course will not be awarded the degree.

(c) CGPA (Cumulative Grade Point Average) is the weighted average of the grade points obtained by a student in all the courses. CGPA will be calculated according to the following formula:

$$\text{CGPA} = \frac{\Sigma(\textit{grade points in a course*credits for the course})}{\textit{total credits taken}}$$

(d) In the tabulation process, only the total marks of a student in any course will be rounded up to next number and the published result of the program will show only the grades earned and the Cumulative Grade Point Average (CGPA) at the end of each semester.

Evaluation of the Courses

The performance of a student in a course will be evaluated in the following ways:

(a) For a theory course the evaluation will be made on the basis of Continuous Evaluation (attendance, quiz/assignment/presentation, mid-term/in-course exam) and Final Exam.

(b) For a lab course, the evaluation will be made on the basis of Continuous Evaluation (attendance, assignment/presentation, lab work, in-course exam) and Lab Final Exam.

(c) For any courses continuous evaluation marks will be evaluated by the course teacher/s and the result must be submitted to the exam committee and controller of the exam before commencement to the semester final examination.

(d) The percentage of attendance of students for all courses along with the attendance sheet must be submitted to the director of the IIT before the announcement of the semester final examination.

(e) The mid-term/in-course exam scripts must be shown to students before the class ends.

(f) If more than one mid-term/in-course exam is conducted, the final mark will be calculated by averaging all of them (the best one will not be allowed).

(g) For theoretical/lab course final exams, there will be two examiners: the course teacher will be the first examiner and the second examiner shall be appointed by the examination committee from the panel of examiners. The second examiner will be within the institute or from a relevant department of the University of Dhaka. If a suitable examiner is not found from the University of Dhaka, a second examiner with prior permission from the Vice Chancellor may be appointed from other universities.

(h) (i) The answer scripts of the final exam will be evaluated by two examiners. The average mark will be considered as the mark obtained if the difference of two examiner marks is less than or equal to 20%.

(ii) In case of a difference in marks between the two examiners is more than 20%, the script in question shall be evaluated by a third examiner. The nearest two marks from the three examiners will be taken for average.

(iii) If more than one set of nearest marks exists then the mark obtained will be calculated from the average of three examiners.

(i) The third examiner for a course shall be appointed by the examination committee from the panel of examiners other than a member of the examination committee or a tabulator, provided further that he/she was not an examiner of this course.

(j) The assessment of the Software Project Lab course will be made by observing the overall performance of a student during practical, viva-voce, assignments, and presentations (as set by the Academic Committee of the institute).

(k) Each final-year project will be supervised by a respective faculty of the Institute. For the fourth-year project, the evaluation will be made on the basis of the supervisor's evaluation, presentation on project defenses, and project report (as set by the Academic Committee of the institute).

(l) For Internship, the evaluation will be made on the basis of assessment by the mentor, presentation, and Internship report (as set by the Academic Committee of the institute).

Exam Committee Formation

(a) At the beginning of each academic semester/session, an exam committee shall be formed for that semester/session by the academic committee of the institute. The Chairman of the exam committee will act as a course coordinator for that semester/session. The role of a course coordinator is to monitor academic activities and report to the director of the institute to avoid any unexpected situation.

(b) The exam committee will consist of four members proposed by the academic committee of the institute.

(c) The committee members are: a chairman, two internal members from the institute and one external member outside of the institute.

(d) In case of any vacancy or inability on the part of a member/examiner, the examination work shall not be invalidated.

(e) The exam committee will be responsible for all exam-related activities as per university rules.

The Marks Distribution for a Course

(a) **For a theoretical course**

i. Continuous Evaluation (attendance, quiz/assignment/presentation, and mid-term/in-course exam; Marks distribution as set by the academic committee)	40%
ii. Final exam	60%
Total Marks	100%

(b) **For a lab course**

i. Continuous Evaluation (attendance, assignment/presentation, in-course exam, and lab work; Marks distribution as set by the academic committee)	70%
ii. Lab final exam	30%
Total Marks	100%

Requirement to Sit for Course Final Exam (Theory and Lab Courses)

(a) Students having 75% or more attendance on average are eligible to appear in the semester final Exam.

(b) Students having average 60-74% attendance will be allowed to sit for the exam with a fine as per university rule.

(c) Students having average attendance below 60% will not be allowed to sit for the semester final exam but may seek readmission in the program.

(d) The duration of course final Exams will be as follows:

Course	Final Exam Marks	Duration of Exam
3 credits theoretical course	60	3 hours
1.5 credits theoretical course	60	2 hours
1.5 credits lab course	30	3 hours
0.75 credits lab course	30	2 hours

(f) In case of a theoretical course (3cr theory), the student will have to answer any 5 (five) questions out of 7 (seven) questions given. Full marks for each question will be 12 (twelve) and the total marks of the course final exams will be (12x5=) 60 (sixty). In case of a theoretical course (1.5cr theory), the student will have to answer any 4 (four) questions out of 6 (six) questions given. Full marks for each question will be 15 (fifteen) and the total marks of the course final exams will be (15x4=) 60 (sixty).

Promotion

(a) A student who has attended courses required for a particular odd semester and appeared at the odd semester's final exams then he/she will be auto-promoted to the next semester.

(b) A student who has attended courses required for a particular even semester, appeared at the even semester's exams and scored a minimum specified CGPA for promotion to the next year, will be promoted to the year.

(c) Promotion to the next year will be given if a student scores minimum CGPA follows:

1st year to 2nd year	: CGPA: 2.00
2nd year to 3rd year	: CGPA: 2.25
3rd year to 4th year	: CGPA: 2.50

Improvement of Grades

(a) Students with 'F' grade in any course, shall get one chance to improve the grade of that course only once with the following batch.

(b) If a student obtains a grade C+ or lower in a course in any year, he/she will be allowed to repeat the term-final examination only once with the following batch for the purpose of grade improvement. A student failing to improve his/her grade in a course can retain the earlier grade. A student will not be allowed for grade improvement if he/she passes and the final semester result is published.

(c) Improvement in 4th year courses: Students would be allowed to sit for improvement examination in the 4th year courses with the following batch, provided they must do it before the publication of the final result by the office of the Controller of Examinations or Issuance of Provisional Certificate by the Controller of Examinations.

(d) Improvement exams will be managed as per University rule.

Readmission

(a) If a student failed to get promotion to the next year then he/she may take readmission with the following batch.

(b) A student can take readmission a maximum of two times throughout the program either in the same class or in different classes. In both cases, he/she must complete the degree by 6 (six) years from the time of original admission.

(c) A student may seek readmission provided he or she has at least 30% (thirty percent) attendance in the previous semester or year.

(d) A student who is unable to get minimum required CGPA even after taking readmission twice will be dropped out from the academic program.

(e) On readmission, grades earned earlier by a student in case of readmission shall, in general, cease to exist and the student has to retake all courses and examinations, but in case they do not get the opportunity to repeat the courses due to late admission, marks of continuous evaluation in the previous year may be retained by the students.

(f) If a student did not appear in the final examination, then he/she will have to seek readmission within a month of the starting of the next academic year. On the other hand, if a student appeared in one or more courses in the final examination, then he/she will have to seek readmission within a month from the date of the publication of the result.

(g) Every application for readmission must be vetted by the academic committee of the institute.

(h) The director of the institute and the concerned hall provost will verify every application as per university practices.

IIT Academic Excellence Award Trust Fund Gold Medal

In recognition of excellent academic performance students may be given the IIT Academic Excellence Award gold medal for every batch after completion of the BSSE program as per the following criteria.

(a) An awardee must not have appeared in any improvement exam during his or her study record.

(b) An awardee must have CGPA of 3.80 or above.

(c) However, the number of awardees of each academic year will be one. In case of equal CGPA, the final semester marks will be considered to break the tie.

Regular Program Office

(a) For smooth administration of the BSSE (and MSSE) program, a Regular Program Office (RPO) shall be formed.

(b) The RPO will consist of a chairperson and all the coordinators of different batches of the BSSE and MSSE programs of the institute.

(c) The Academic committee of the institute shall nominate the chairperson of the RPO from its current member

(d) The Academic committee of the institute shall also nominate one of its members to act as the coordinator for each academic year of the BSSE Program (and each batch of MSSE Program).

(e) The RPO committee will assist the Director of the institute in matters relating to the improvement of grades, readmission, promotion, Class routine, exam scheduling, preparation of IIT Gold Medal, monitor the academic activities and report to the director of the institute to avoid any unexpected situation.

(f) The coordinator for each academic year of the BSSE Program (and each batch of MSSE Program) will act as the chairman of the exam committee for the semesters of that academic year (or batch).

(g) The tenure of the committee will be one academic year.

Other General Regulations

(a) For any matter not covered in the above guidelines, existing rules of the University of Dhaka will be applicable.

(b) Disciplinary and punishable actions will be applied according to the existing rules of the University of Dhaka.

Semester-Wise Courses

Semester 1

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
CSE 1101	Structured Programming	3	3	0
CSE 1102	Discrete Mathematics	3	3	0
STAT 1103	Probability and Statistics for Engineers I	3	3	0
MATH 1107	Calculus I	3	3	0
GE 1105	Introduction to Sociology	3	3	0
SE 1106	Introduction to Software Engineering	3	3	0
CSE 1101L	Structured Programming Lab	1.5	0	1.5
Total Credits		**19.5**	**18**	**1.5**

Semester 2

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
CSE 1215	Data Structure	1.5	1.5	0
MATH 1216	Calculus II	3	3	0
CSE 1211	Computer Organization	3	3	0
STAT 1203	Probability and Statistics for Engineers II	3	3	0
GE 1212	Bangladesh Studies	3	3	0
SE 1206	Object Oriented Concepts	3	3	0
CSE 1215L	Data Structure Lab	1.5	0	1.5
CSE 1211L	Computer Organization Lab	0.75	0	0.75
SE 1206L	Object Oriented Concepts Lab	1.5	0	1.5
Total Credits		**20.25**	**16.5**	**3.75**

Semester 3

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
CSE 2115	Introduction to Algorithm	3	3	0
SE 2112	Theory of Computing	3	3	0
CSE 2111	Computer Networking	3	3	0
MATH 2116	Linear Algebra	3	3	0
SE 2105	Software Project Lab I	3	0	3
BUS 2117	Business Communication	3	3	0
CSE 2115L	Introduction to Algorithm Lab	1.5	0	1.5
CSE 2111L	Computer Networking Lab	0.75	0	0.75
Total Credits		**20.25**	**15**	**5.25**

Semester 4

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
CSE 2201	Operating System and System Programming	3	3	0
SE 2215	Design Patterns	1.5	1.5	0
CSE 2204	Database Management System	3	3	0
BUS 2205	Business Studies for Engineers	3	3	0
SE 2206	Software Requirements Specification and Analysis	1.5	1.5	0
CSE 2201L	Operating System and System Programming Lab	1.5	0	1.5
SE 2215L	Design Patterns Lab	1.5	0	1.5
CSE 2204L	Database Management System Lab	1.5	0	1.5
SE 2206L	Software Requirements Specification and Analysis Lab	1.5	0	1.5
Total Credits		**18**	**12**	**6**

Semester 5

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
SE 3115	Software Testing and Quality Assurance	1.5	1.5	0
CSE 3102	Web Technology	1.5	1.5	0
CSE 3116	Mathematical and Statistical Methods for Data Analysis	3	3	0
CSE 3117	Artificial Intelligence	3	3	0
SE 3118	Software Design and Analysis	1.5	1.5	0
CSE 3119	Cryptography and Security Mechanisms	3	3	0
SE 3105	Software Project Lab II	3	0	3
CSE 3102L	Web Technology Lab	1.5	0	1.5
CSE 3119L	Cryptography and Security Mechanisms Lab	0.75	0	0.75
Total Credits		**18.75**	**13.5**	**5.25**

Semester 6

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
SE 3215	Internship	15	0	15
Total Credits		**15**	**0**	**15**

Semester 7

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
CSE 4115	Distributed Systems	1.5	1.5	0
CSE 4116	Compiler Design	3	3	0
SE 4117	IT Laws & Ethics	3	3	0
CSE/SE 41XX	Elective 1	3	3	0
CSE 4118	Machine Learning	3	3	0
CSE 4115L	Distributed Systems Lab	1.5	0	1.5
SE 4119	Project	3	0	3
Total Credits		**18**	**13.5**	**4.5**

Semester 8

Course Code	*Course Title*	*Credit*	*Theory*	*Lab*
SE 4215	Software Security	1.5	1.5	0
SE 4203	Software Project Management	3	3	0
CSE/SE 42XX	Elective 2	3	3	0
CSE/SE 42XX	Elective 3	3	3	0
SE 4201	Project	3	0	3
SE 4215L	Software Security Lab	1.5	0	1.5
SE 4216	Software Metrics	1.5	1.5	0
SE 4216L	Software Metrics Lab	0.75	0	0.75
Total Credits		**17.25**	**12**	**5.25**

Elective Courses

No.	*Track*	*Course Code*	*Course Title*
1	**Data Science**	CSE 4150	Information Retrieval
		CSE 4250 CSE 4251	Natural Language Processing Introduction to Data Mining
2	**Security**	CSE 4160	Network and System Security
		CSE 4261 CSE 4262	Digital Forensics Cyber Physical System Security
3	**Computer Vision and Image Processing**	CSE 4170	Pattern Recognition
		CSE 4270 CSE 4271	Image Processing Photogrammetry
4	**Bioinformatics**	CSE 4280 CSE 4281	Introduction to Bioinformatics Genomic Data Science
5	**Advanced Software Engineering**	SE 4290 SE 4291 SE 4292	Software Maintenance Human Computer Interaction Service-Oriented Architecture

PART C: Description of All Courses

Semester 1 (1^{st} year 1^{st} Semester)

Course No.	Course Title	No. of Credits	Credit Hours
CSE 1101	**Structured Programming**	**Theory: 3**	15 × (1×3) = 45

Introduction of the Course: This course, which treats problem solving as a fundamental subject, gives students both theoretical and practical experience in this area. To tackle easy to difficult problems, students are expected to write programs in the structured programming language. Preprocessor directives, arrays, constants, variables, data types, input and output instructions, text files, and control structures are all topics covered in the course. Additionally, basic library functions might be included.

Specific Objectives:

- To design and develop solution approach of any programming problem following the structured programming paradigm
- To apply structured programming concepts to solve real-life problems

Course Contents: Fundamentals of C programming; Introducing C's Program Control Statements; Data types, Variables and Expressions; Exploring Arrays and Strings; Understanding Pointers and Functions; Console and File I/O; Structures and Unions.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. Teach Yourself C, Herbert Schildt, McGraw Hill
2. C: The Complete Reference, Herbert Schildt, McGraw Hill
3. Schaum's Outline of programming with C, McGraw Hill

Course No.	Course Title	No. of Credits	Credit Hours
CSE 1102	**Discrete Mathematics**	**Theory: 3**	15 × (1×3) = 45

Introduction of the Course:

Discrete mathematics is the part of mathematics devoted to the study of discrete objects. Discrete mathematics is used whenever objects are counted, when relationships between finite sets are studied, and when processes involving a finite number of steps are analyzed. Since information is stored and manipulated by computing machines in a discrete fashion, Discrete Mathematics is the backbone of Computer Science.

Specific Objectives:

- To understand, construct and prove mathematical arguments
- To develop recursive algorithms based on mathematical induction
- To know basic properties of functions and relations
- To know essential concepts in graph theory and related algorithms
- To understand concepts of counting using permutation and combination
- To understand basics of number theory and cryptography
- To apply knowledge about discrete mathematics in problem solving

Course Contents:

Logic and Proofs: propositional logic, applications of propositional logic, propositional equivalences, predicates and quantifiers, nested quantifiers, rules of inference, introduction to proofs.

Induction and Recursion: Mathematical Induction, Recursive Definitions and Structural Induction, Strong induction.

Basic Structures: Sets, Functions, Sequences, Sums, Matrices.

Relations: Symmetry, transitivity, reflexivity, representing relations, closure of relations, equivalence classes, partial orderings.

Graphs: Graphs and graph models, graph terminology, representing graphs, graph isomorphism, connectivity, graph coloring, Euler and Hamilton paths.

Trees: Introduction and application of trees, introduction to binary search tree and decision tree, tree traversal, spanning trees.

Counting: The addition and multiplication rules, The principle of Inclusion-Exclusion, The pigeon-hole principle, permutations, combinations, Generalized Permutations and Combinations.

Number Theory: The division algorithm, divisibility and the Euclidean algorithm, prime numbers, congruence, applications of congruence;

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Real world problem solving in groups

Prerequisite: N/A

References:

1. Discrete Mathematics and Its Applications, 8th Ed. – Kenneth H. Rosen
2. Discrete Mathematics: An Open Introduction, 3rd Ed. – Oscar Levin
3. Discrete Mathematics with Applications, 4th Ed. – Susanna S. Epp

Course No.	Course Title	No. of Credits	Credit Hours
STAT 1103	**Probability and Statistics for Engineers I**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course: This course will explain the basics of probability and statistics for engineers. It will cover the Introduction to Statistics, Descriptive Statistics, Elements of Probability, Random Variables and Expectations, Special Random Variables, Distributions of Sampling Statistics, and Parameter Estimation. It is expected that, by the end of this course, students will be able to describe how probability and statistics will be used for engineering activities.

Specific Objectives:

- To understand the basics of probability and statistics
- To understand the statistics, elements of probability, random variables and expectation
- To describe how special random variables, distributions of sampling statistics, and parameter estimation work

Course Contents: Introduction to Statistics: Concept of Data and Variables, Data Collection and Descriptive Statistics, Inferential Statistics; Populations and Samples; Descriptive Statistics: Frequency Tables and Graphs, Relative Frequency Tables and Graphs, Grouped Data, Histograms, Ogives, Stem and Leaf Plots, Sample Mean, Sample Median, Sample Mode, Sample Variance and Standard Deviation, Sample Percentiles and Box Plots, Chebyshev's Inequality, Normal Data Sets, Paired Data Set and Sample Correlation Coefficient; Elements of Probability: Basic Terminology in Probability, Sample Space and Events, Venn Diagrams and Algebra of Events, Axioms of Probability, Conditional Probability, Bayes' Theorem and Independent Events; Random Variables and Expectation: Random Variables, Types of Random Variables, Jointly Distributed Random Variables, Expectation, Property of Expected Values, Use of Expected Values in Decision Making, Variance, Covariance and Variance of Sums of Random Variables and Moment Generating Functions; Special Random Variables: Binomial Random Variables, Poisson Random Variables, Uniform Random Variables, Normal Random Variables, Exponential Variables, Gamma Distribution, Chi-Square Distribution, t-Distribution and F-Distribution; Distributions of Sampling Statistics: Central Limit Theorem, Sampling Distribution for Normal Population, and Sampling from a Finite Population; Parameter Estimation: Maximum Likelihood Estimators, Interval Estimates, Estimating the difference in Means of Two Normal Population, Approximate Confidence Interval for the Mean, Confidence Interval of the Mean of the Exponential Distribution and Bayes' Estimator.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Elsevier/Academic Press, 4th Ed.
2. Hayter A, Probability and Statistics for Engineers and Scientists, Cengage Learning, 4th edition

Course No.	Course Title	No. of Credits	Credit Hours
MATH 1107	**Calculus I**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course:

Ideas of calculus include rates of change of a function, which is known as the derivative of a function, and integrals which is the reverse process of derivatives. This course module concentrates on in-depth understanding of these two concepts and their properties to communicate the results in applications of Calculus in science and engineering through modeling natural phenomena.

Specific Objectives:

- To explain the general concept of functions and sketching their graphs
- To explain concepts of the limits, continuity, and derivative
- To calculate derivatives for various types of functions
- To apply the concept of derivative to completely analyze a function and various other applications of the derivative
- To compute antiderivative of functions and apply the Fundamental Theorem of Calculus

Course Contents:

Calculus of single variable functions: Functions, Graphs and Algebra of Functions, Limit and continuity, Rules of Finding Limits; The Derivatives of Functions: Tangent line, Local Linear approximation and differential, Differentiation Rules, Chain Rule, Implicit Differentiation, Related Rates of Change, Mean Value Theorem, Partial derivatives, Taylor series with Remainder, Indeterminate forms and L'Hopital's rule, Interval of increasing/decreasing, concavity, Extreme Values of Functions, First Derivative and Second Derivative Tests for Extreme Values, Optimization; Integrals: Indefinite Integrals, Integration by Parts, Integration by Substitution, Integration by Partial fractions, Riemann Sums, Definite Integral, Fundamental Theorem of Calculus, Average value of a function and Mean Value Theorem of integration, Substitution in Definite Integrals, Areas between Curves, Surface area and Volume of the surface of Revolution, Lengths of Plane Curves, Improper Integrals, Infinite series and convergence tests. Numerical Methods: Bisection, False Position, Newton-Raphson, Secant method, Fixed point iteration methods for approximating solutions of nonlinear equations, Newton's method for approximating solutions of systems of nonlinear equations.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. Howard Anton, Calculus, John Willey & Sons, Inc.
2. G.B. Thomas and R.L. Finney, *Calculus and Analytical Geometry*, Addison Wesley, 9th Ed.
3. Erwin Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons, 9th Ed.

Course No.	Course Title	No. of Credits	Credit Hours
GE 1105	**Introduction to Sociology**	**Theory: 3**	15 × (1×3) = 45

Introduction of the Course: Sociology is the scientific study of society. It is an overarching unification of all studies of human society, including philosophy, economics, anthropology, history, political science, and women and gender studies. It borrows ideas from mathematics, physics, chemistry, psychology, and biology to understand society. The purpose of this course is to introduce students to the discipline through critical readings and discussions. The course will prepare students to understand some fundamental questions. Why do we live in a society? How do we create our meanings in everyday life? How can we understand Bangladeshi society and global societies? Who dominates our society and oppresses the marginalized communities? Why do poverty, racism, ethnocentrism, sexism, classism, and other forms of inequalities exist in society? How can we deal with such social hierarchies and contribute to social policy towards a just society? Sociologists pose these questions and offer potential answers using sociological theories and methods and providing historical and empirical evidence.

Specific Objective:

- To provide students with tools for understanding their own social position and the conditions in which they live
- To fuel students passion and vision for a just, equal, peaceful and diverse society
- To understand the surroundings of society and their interactions
- To learn how to develop and conduct a social research in a practical field
- To know how to collect, present and analyze social research findings

Course Contents:

Foundation of Sociology: Defining society and sociology. Sociology as a science. Origin and development of sociology as an academic discipline: Enlightenment, Industrial Revolution, French Revolution, and The United States Bill of Rights.

Theoretical Approaches to Sociology: Major sociological perspectives: positivist, conflict, interactionist, critical, and postmodernist. Classical Theorists: August Comte, Karl Marx, Emile Durkheim, Max Weber and Du Bois. Contemporary Theorists: Antonio Gramsci, Herbert Marcuse, Charles W Mills, Pierre Bourdieu, Zygmunt Bauman, Manuel Castells, and Anthony Giddens.

Sociological Research Methods: Defining scientific research in sociology. Research methodologies: qualitative (interview and ethnography) and quantitative (survey). Basic steps of scientific sociological research: defining a research problem, reviewing existing literature, developing research questions or formulating hypotheses, designing research, collecting data, analyzing findings (using SPSS for quantitative data and NVivo for qualitative data), and writing a report.

Society, Social Institutions and Social Organizations: Evolution of modern human civilization. Types of society: ancient, feudal, and capitalist. Types of social institutions: marriage, family, education, and religion. Forms of social organizations: state, party, market, class, and status groups.

Social Stratifications and Social Problems: Forms of social inequalities- slavery, estate, race, ethnicity, class, caste, gender, and sex. Social problems: war, crime, violence, ethnic conflict, drug trafficking, human trafficking, drug abuse, delinquency, and terrorism; displacement, dispossession, poverty, famine, malnutrition, disabilities, marginality, exclusion, and illiteracy.

Social and Environmental Changes: Forms of globalization- economic, political, social and cultural. Modernization of agriculture, urbanization, industrialization, migration, and remittance. Digital civilization: fourth and fifth industrial revolutions, information technology, and network society. Environmental crises: greenhouse effect, carbon emission, deforestation, drought, sea-level rise, salinity, flood, and cyclones; water and air pollution, soil erosion, and other environmental degradations.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. Sociology, 16th Edition, 2017 (Global Edition), by John J. Macionis.

Course No.	Course Title	No. of Credits	Credit Hours
SE 1106	**Introduction to Software Engineering**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course:

This course is an introduction to Software Engineering. The main objective of the course is to know the basic concepts of software engineering, information science and information technology. By learning this course the students will be able to understand the different umbrella activities of software engineering where process management and measurement are involved.

Specific Objectives:

- To get the basic understanding of software engineering
- To understand the system process behavior
- To understand the management and different software paradigm activities
- To analyze software measurement and measuring techniques
- To get the basic understanding of software requirements specification
- To get the basic understanding of software risk, software testing and software quality

Course Contents: Introduction; Overview of Software Engineering: history, nature, relation of software engineering to other disciplines, software development life cycle, Programming language. Software nature and qualities: product qualities, project qualities, correctness, robustness, usability, maintainability, portability, quality measurements; Software development life cycle: requirement, design, development, testing, maintenance; Software development models: waterfall, agile, spiral, RDD, V model. Software measurement: modularity, abstraction, generality, object oriented, component oriented, structured. Specification and Verification: requirement specification, descriptive specification, testing, analysis, debugging; Modeling and Design: basics of modeling diagram, UI design. Software Project Management: concepts project metrics, estimation risks management.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss theoretical concepts in a real time project (using a project concept).

Prerequisite: N/A

References:

1. Software Engineering: A Practitioner's Approach, 7th Edition, McGraw Hill Higher Education.
2. Software Engineering, 7th edition Sommerville.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 1101L	**Structured Programming Lab**	1.5	**15 × (1.5×2) = 45**

Course Contents: Implementation and relevant lab work of the corresponding theory course.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. Software Engineering: A Practitioner's Approach, 7th Edition, McGraw Hill Higher Education.
2. Software Engineering, 7th edition, Sommerville.

Semester 2 (1^{st} year 2^{nd} Semester)

Course No.	Course Title	No. of Credits	Credit Hours
CSE 1215	**Data Structure**	**Theory: 1.5**	**15 × (1.5×1) = 22.5**

Introduction of the Course: This course will illustrate the topics of data structures and algorithms. The course will guide the students to use these concepts in real life problems. It is expected that, by the end of this course, students will be able to use the most suitable data structure in different practical scenarios using this knowledge.

Specific Objectives:

- To solve real life problems using data structures and algorithms
- To allow easier processing of data
- To design efficient algorithms
- To visualize data in graphical view
- To store data efficiently and effectively
- To increase problem solving attitude
- To learn about many searching and sorting strategies

Course Contents: Data Structures and Complexity of Algorithms, Time Space Tradeoff, Searching Techniques: Linear and Binary Searching; Sorting and Recursion - Discussion of Common Sorting Techniques: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort, Merge Sort, Radix Sort; Factorial and Tower of Hanoi Problem; Linked Lists - Abstract Data Types (ADT), List ADTs, and Linked Lists: Singly, Two Way and Circular Linked Lists; Stacks and Queues - Stacks and Queues and their Implementation Strategies;

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1101, CSE 1102.

References:

1. *Data Structures*. Schaum's Outline Series Effective Java, 2nd Edition, Joshua Bloch, Addison-Wesley
2. E. Horowitz and S. Sahni, *Fundamentals of Data Structures*, London Pitman
3. Robert L. Kruse, *Data Structures and Program Design*, Prentice Hall, 2nd Ed.

Course No.	Course Title	No. of Credits	Credit Hours
MATH 1216	**Calculus II**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course:

Ideas of calculus include rates of change of a function which is known as the derivative of a function and integrals which is, in particular, reverse process of derivatives. This course module concentrates on in depth understanding of these two concepts and their properties for functions of more than one variable to communicate the results in applications of Calculus in science, engineering through modeling natural phenomena.

Specific Objectives:

- To gain familiarity with key ideas of precalculus, including the manipulation of equations and elementary functions
- To develop fluency with the preliminary methodology of tangents and limits, and the definition of a derivative
- To develop and practice methods of differential calculus with applications
- To develop and practice methods of the integral calculus

Course Contents:

Calculus of multivariable variable functions (mostly two variables).

Functions, Graphs of Functions, Limit and continuity, Rules of Finding Limits, The Partial Derivatives of a Function, Tangent plane, Local Linear approximation and differential, Chain Rules, Related Rates of Change, Taylor series with Remainder, Extreme Values of Functions, First Derivative and Second Derivative Tests for Extreme Values, Optimization, Vectors in space, Dot and Cross products, Lines and Planes in Space, Lagrange multiplier technique for Constrained optimization. Gradient Descent method, Steepest Descent Method, Multiple Integrals with some applications.

Differential Equations and Mathematical Modeling: First Order Initial Value Problems, Homogeneous Linear Equations with Constant Coefficients, Non-homogeneous Equations and Variation of Parameters. Critical Points, Stability of critical points, Qualitative Methods for Linear and Nonlinear Linear System of ordinary differential equations. Numerical Methods: Euler Method, Range-Kutta Method of order two and four for IVP.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: MATH 1107

References:

1. S.L. Ross, Differential Equation.
2. D.G.Zill, A First Course in Differential Equations with Applications.
3. Numerical Methods, E Balagurusamy, Tata McGraw-Hill Publishing Company.
4. Numerical Methods, Rao V. Dukkipati, New Age International Ltd.
5. Numerical Analysis, Richard L. Burden & J. Douglas Faires, Cengage Learning.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 1211	**Computer Organization**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course: This course introduces the structure and behavior of the computer systems to the students. Topics include the functional components of a computer, performance computation and performance issues, instruction sets and instruction set architectures, memory, device, multithreading and multiprocessing. This course will enable the students to apply the aforementioned concepts to understand, analyze and solve recurring problems of a computer system.

Specific Objectives:

- To estimate a modern digital computer's performance based on different algorithms, processor speed, and cycles per instruction.
- To demonstrate a thorough comprehension of the RISC processor's instruction set.
- To describe fixed-point arithmetic hardware algorithms.
- To demonstrate thorough comprehension of data and control flow in a contemporary RISC processor.
- To demonstrate a thorough understanding of the various memory management and organization techniques.
- To determine the necessary size of a cache based on the size of the index, the size of the tag, and the associativity.
- To calculate the performance of different types of memory based on specified criteria

Course Contents: Function and structure of a computer, Functional components of a computer, Interconnection of components, Performance of a computer; Number representation and arithmetic: Binary, octal, and hexadecimal numbers, One's and two's complements and other representations, Addition and subtraction; Digital logic and integrated circuits: Boolean algebra and truth tables, Boolean functions (Gates, Functions, Simplification), Integrated circuits (Combinational circuits - adders, shifters, decoders, multiplexers and ROM's; Flip-flops; Sequential circuits - registers, counters and RAM); Representation of Instructions: Machine instructions, Operands, Addressing modes, Instruction formats, Instruction sets, Instruction set architectures - CISC and RISC architectures; Introduction to Assembly Language: Programming with Assembly language, The assembly process, Linking and loading, Register-level debugging, Processing Unit: Organization of a processor - Registers, ALU and Control unit, Data path in a CPU, Instruction cycle, Organization of a control unit - Operations of a control unit, Hardwired control unit, Micro-programmed control unit; Memory Subsystem: Semiconductor memories, Memory cells - SRAM and DRAM cells, Internal Organization of a memory chip, Organization of a memory unit, Error correction memories, Interleaved memories, Cache memory unit - Concept of cache memory, Mapping methods, Organization of a cache memory unit, Fetch and write mechanisms, Memory management unit - Concept of virtual memory, Address translation, Hardware support for memory management; Input/Output Subsystem: Access of I/O devices, I/O ports, I/O control mechanisms - Program controlled I/O, Interrupt controlled I/O, and DMA controlled I/O, I/O interfaces - Serial port, Parallel port, PCI bus, SCSI bus, USB bus, Firewall and InfiniBand, I/O peripherals - Input devices, Output devices, Secondary storage devices; Multiprocessing Systems: Shared memory multiprocessor, Message-passing multiprocessor, Hardware multithreading

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. D. A. Patterson and J. L. Hennessy, "Computer Organization and Design - The Hardware/Software Interface", Morgan Kaufmann, 1998.
2. 2. C. Hamacher, Z. Vranesic and S. Zaky, "Computer Organization", McGraw-Hill, 2002.
3. 3. W. Stallings, "Computer Organization and Architecture - Designing for Performance", Prentice Hall of India, 2002.
4. 4. J .P. Hayes, "Computer Architecture and Organization", McGraw-Hill, 1998.

Course No.	Course Title	No. of Credits	Credit Hours
STAT 1203	**Probability and Statistics for Engineers II**	**Theory: 3**	15 × (3×1) = 45

Introduction of the Course:

This course will explain the advances of probability and statistics for engineers. It will cover Hypothesis Testing, Regression and Correlation Analysis, Analysis of Variance, Goodness of Fit Tests and Categorical Data Analysis, Nonparametric Hypothesis Tests, and Quality Control. It is expected that, by the end of this course, students will be able to describe how probability and statistics will be used for advanced engineering activities.

Specific Objectives:

- To understand the advances of probability and statistics
- To understand hypothesis testing, regression and correlation analysis, and analysis of variance
- To describe how goodness of fit tests and categorical data analysis, nonparametric hypothesis tests, and quality control works

Course Contents:

Hypothesis Testing: Tests Concerning the Mean of a Normal Population, Testing the Equality of Means of Two Normal Populations, Hypothesis Tests Concerning the Variance of a Normal Population, Hypothesis Tests in Bernoulli Populations and Tests Concerning the Mean of a Poisson Distribution. **Regression and Correlation Analysis:** Least Squares Estimators of the Regression Parameters, Distribution of the Estimators, Statistical Inference about the Regression Parameters, Coefficient of Determination and Sample Correlation Coefficient, Analysis of Residuals, Transforming to Linearity, Weighted Least Squares, Polynomial Regression, Multiple Linear Regression, Logistic Regression Models for Binary Output Data and Correlation Analysis. **Analysis of Variance:** One-way Analysis of Variance, Two-Factor Analysis of Variance: Introduction and Parameter Estimation, Testing Hypotheses, and Two-way Analysis of Variance with Interaction Problems. **The goodness of Fit Tests and Categorical Data Analysis:** Goodness of Fit Tests when All Parameters are Specified, Goodness of Fit Tests when All Parameters are Unspecified, Tests of Independence in Contingency Tables, Tests of Independence in Contingency Tables Having Fixed Marginal Totals and Kolmogorov-Smirnov Goodness of Fit Test for Continuous Data. **Nonparametric Hypothesis Tests:** Sign Test, Signed Ranked Test, Two-Sample Problem, and Runs Tests for

Randomness. **Simulation, bootstrap statistical methods, and permutation tests:** Random numbers. The bootstrap methods, permutation tests. Generating discrete and continuous random variables.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: STAT 1103

References:

1. Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Elsevier/Academic Press, 4th Ed.
2. Douglas C. Montgomery and George C. Runger, Applied Statistics and Probability for Engineers, John Wiley and Son, 4th Ed.

Course No.	Course Title	No. of Credits	Credit Hours
GE 1212	**Bangladesh Studies**	**Theory: 3**	15 × (3×1) = 45

Introduction of the Course:

This course will explain the History and Society of Bangladesh. It will cover the Government and Politics, Geography and Resources of Bangladesh, Social Structure of Bangladesh, Culture of Bangladesh, Economy of Bangladesh, Achievements in different sectors of Bangladesh, and Socio-economic problems and prospects of Bangladesh. It is expected that, by the end of this course, students will be able to understand Bangladeshi history, society, culture, overall economics and others.

Specific Objectives:

- To understand the History and Society of Bangladesh
- To understand the Government and Politics, Geography and Resources of Bangladesh, Social Structure of Bangladesh, Culture of Bangladesh, Economy of Bangladesh
- To understand the Achievements in different sectors of Bangladesh, and Socio-economic problems and prospects of Bangladesh

Course Contents: History and Society of Bengal under the British rule and Pakistan rule: The impact of British and Pakistan rules on the economy and education of the people. Language Movement of 1952, Events Leading to the Mass Upsurge of 1969, War of Independence and the Emergence of Bangladesh in 1971, **Government and Politics:** Three branches of government-executive, legislative and judiciary, Formation and role of major political parties in Bangladesh and Constitutional development of Bangladesh, **Study of Geography and Resources of Bangladesh:** Location, Area, Boundary, Ecological Settings, River System, Climate, People and Resources of Bangladesh, **Social Structure of Bangladesh:** Rural society, Urban society, Family, household, and kinship, Women's role in society **Culture of Bangladesh:** Language, Literature, Art and Culture of Bangladesh, **Economy of Bangladesh:** Major Economic Sectors, potentials of various sectors and their prospective challenges; foreign aid and development, role of donor agencies, role of NGOs, **Achievements in different sectors of Bangladesh:** Economy, Culture, Sports, etc., **Socio-economic problems and prospects of Bangladesh:** poverty, health issues, natural disaster, social stratification and gender discrimination.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: N/A

References:

1. Islam, S. (2003). Banglapedia. National Encyclopedia, Asiatic Society of Bangladesh, Dhaka
2. Kibria, S. A. (1999). *Bangladesh at the Crossroads*. University Press Ltd.
3. Riaz, A. (2016). Bangladesh: A Political History since Independence. London: IB Taurus.

Course No.	Course Title	No. of Credits	Credit Hours
SE 1206	**Object Oriented Concepts**	**Theory: 3**	15 × (3×1) = 45

Introduction of the Course:

This course will explain the basic topics of object oriented concepts. The course will guide the students to design a problem and code them using Object Oriented (OO) Concepts. It is expected that, by the end of this course, students will be able to model different practical scenarios using the object oriented paradigm. Students will learn and implement concepts, functionalities and interaction of objects and classes using different programming languages, which will strictly follow Object Orientation.

Specific Objective:

- To understand the principle of Object Oriented Architecture
- To design problems from the perspective of Object Oriented Architecture
- To be able to manage data, functions and user interactions using Object Orientation
- To handle interconnection activities among different objects
- To handle multithreaded activity on different objects

Course Contents:

Introduction to Object Oriented Concepts: Procedural vs Object Oriented (OO) Programming, What is an Object – Object Data and Behavior, What is a Class – Attributes, Methods and Messages; **The Anatomy of a Class**: The Name, Comments, Attributes, Constructors, Accessors, Public Interface methods and Private implementation methods; **Constructors:** Default constructor, Using multiple constructors; **Object Oriented principles:** Encapsulation, Inheritance, and polymorphism; **Encapsulation:** Data Hiding, Interfaces and Implementations, **Inheritance:** Superclasses and Subclasses, Abstraction and Is-a Relationships, Reusing Objects, Generalization and Specialization, How Inheritance Weakens Encapsulation; Polymorphism, **Composition:** Abstraction and Has-a Relationships; How to think in terms of Objects – Interface vs Implementation, Abstract thinking when designing Interfaces and Giving the user minimal Interface possible; **Object Oriented Design Principle:** SOLID - Single Responsibility Principle, Open/Close Principle, Liskov Substitution Principle, Interface Segregation Principle and Dependency Inversion Principle.; **User Interface Handling:** User Interface Design, Layout, Item, Mouse Listener, Event Listener, etc. ; **Data Storage:** Store Data (Variable and Objects) in File and Database; Exception Handling; Object Serialization and Deserialization; **Collections and Generics:** Objects in various collections such as List, Map, Vector, Tree, Dictionary, etc.; **Object and Threading:** Thread, Multithreading, Forking, Concurrency, Sleep, etc.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, source codes and documents

Prerequisite: CSE 1101

References:

1. The Object-Oriented Thought Process (4th Edition) Matt Weisfeld, Addison-Wesley, 2013
2. Java: The Complete Reference, Ninth Edition Herbert Schildt, McGraw Hill
3. Java: How to Program, Eleventh Edition Paul Deitel, Harvey Deitel, Pearson, 2017
4. Effective Java, Third Edition, Joshua Bloch, Addison-Wesley

Course No.	Course Title	No. of Credits	Credit Hours
CSE 1215L	**Data Structure Lab**	**Lab: 1.5**	15 × (1.5×2) = 45

Course Contents: Review of data structures and algorithms: Solving problems (described scenarios) using data structures and algorithms in different languages such as C++, Java, Python, C# etc.; Implementing best searching techniques (for some specified scenarios): Linear and Binary Searching; Sorting and Recursion - Solving problems using common techniques: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort, Merge Sort, Radix Sort; Implementing Factorial, Tower of Hanoi and similar type problems; Problem solving using data structures: Linked Lists - Abstract Data Types, List ADTs, and Linked Lists: Singly, Two Way and Circular Linked Lists; Stacks and Queues - Stacks and Queues and their Implementation Strategies.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1101, CSE 1102

References:

1. *Data Structures*. Schaum's Outline Series Effective Java, 2nd Edition, Joshua Bloch. Addison-Wesley
2. E. Horowitz and S. Sahni, *Fundamentals of Data Structures*, London Pitman
3. Robert L. Kruse, *Data Structures and Program Design*, Prentice Hall, 2nd Ed.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 1211L	**Computer Organization Lab**	**Lab: 0.75**	15 × (0.75×2) = 22.5

Course Contents: Implementation and relevant lab work of the corresponding theory course.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. D. A. Patterson and J. L. Hennessy, "Computer Organization and Design - The Hardware/Software Interface", Morgan Kaufmann, 1998.
2. C. Hamacher, Z. Vranesic and S. Zaky, "Computer Organization", McGraw-Hill, 2002.
3. W. Stallings, "Computer Organization and Architecture - Designing for Performance", Prentice Hall of India, 2002.
4. J .P. Hayes, "Computer Architecture and Organization", McGraw-Hill, 1998.

Course No.	Course Title	No. of Credits	Credit Hours
SE 1206L	**Object Oriented Concepts Lab**	**Lab: 1.5**	**15 × (1.5×2) = 45**

Course Contents:

The lab part will be designed based on any Object Oriented Programming Language. In each lab class, students will be given a scenario/ task where the realization of Object Oriented Concepts along with OO Language syntax will be implemented. Lab topics are given below.

Introduction and Environment Preparation: Introduction and installation of OO Language, Integrated Development Environment (IDE); Program compiling and running OO program, using classpath and Architecture;
Syntax – Package, Import, Class, Fields, Methods, Constructors, Primitive data types, condition, iteration, Strings and literal, Nonexistence type: null. Class type: normal, inline, abstract, interface, etc. Programming features to support practical OOP – String Operations: String creations and operations, immutability property of String, String comparison and searching, String buffers and builders; Object and data I/O: Streams, Input and Output Stream, File, Path, Directory and tree; Exception handling: try and catch, checked exception vs unchecked exceptions, throw and throws, Common exception and User defined exceptions; Logger and Debugging: Logger, Log levels, Formatters and Filters, Logger Handlers and Manager; Generics: Wildcard, Generic class definitions, Generic method definitions, Using generics; Collection Framework: Collection interfaces, List and SortedList, Map and SortedMap, Navigable Map, Set and Sorted Set, Navigable Set, Queue and DeQueue, Stack; Data Storage: save in file, CRUD operation in any SQL database; Implementing User Interface; Multi-Threaded Programming: Overview of Thread, Creating and Running Thread, Thread Pools, Thread Synchronization, wait and notify, join and sleep and The concurrency API; Object Serialization and Deserialization.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, source codes and documents

Prerequisite: CSE 1101

References:

1. The Object-Oriented Thought Process (4th Edition) Matt Weisfeld, Addison-Wesley, 2013
2. Java: The Complete Reference, Ninth Edition Herbert Schildt, McGraw Hill
3. Java: How to Program, Eleventh Edition Paul Deitel, Harvey Deitel, Pearson, 2017
4. Effective Java, Third Edition, Joshua Bloch, Addison-Wesley.

Semester 3 (2^{nd} year 1^{st} Semester)

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2115	**Introduction to Algorithm**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course: This course will help the students to analyze and design algorithms. The course will guide the students to use these concepts in real life problems. It is expected that, by the end of this course, students will be able to use best fitted algorithm in different practical scenarios using this knowledge. It is also expected that, students will know which algorithms and techniques are best in which situation and the computational expense while solving any practical problems.

Specific Objectives:

- To allow easier processing of data
- To design an efficient algorithm
- To analyze the computation time of algorithm
- To make a tradeoff between space and time while algorithm selection
- To increase problem solving attitude
- To learn about optimizing problems and its solution procedure

Course Contents: Introduction - Algorithms, Analyzing & Designing Algorithms, Correctness of Algorithms; Greedy Algorithms - Introduction to Greedy Algorithms, Greedy Choice Property, Greedy vs. Dynamic Programming, Fractional Knapsack Problem, Activity Selection Problem, Huffman Encoding, Task Scheduling Problem, Coin Changing Problem, Kruskal's and Prim's Minimum Spanning Tree Algorithms; Divide and Conquer Algorithms - Introduction to Divide and Conquer Design Technique, Quick Sort, Merge Sort, Proof of Correctness, and Run Time Analysis; Dynamic Programming - Introduction to Dynamic Programming Technique, Principle of Optimality, Optimal Substructure Property, Assembly Line Scheduling, Matrix Chain Multiplication, LCS, Viterbi Algorithm, Bitonic Euclidean Traveling Salesperson Problem and Runtime Analysis; Graph Searching and Shortest Path Problems - Breadth First Search, Depth First Search, Flow Networks, Single Source and All Pair Shortest Path Algorithms; Selected Topics - String Matching Algorithms; NP Completeness and Approximation Algorithms - NP Completeness, Polynomial Time Verification, NP Completeness and Reducibility, NP Complete Problems and Approximation Algorithms.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1215, SE 1206

References:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.

Course No.	Course Title	No. of Credits	Credit Hours
SE 2112	**Theory of Computing**	**Theory: 3**	15 × (1×3) = 45

Introduction of the Course:

This course introduces the foundations of automata theory, computability theory, and complexity theory. It shows the relationship between automata and formal languages and addresses the issue of which problems can be solved by computational means (decidability vs undecidability), and introduces concepts related to computational complexity of problems.

Specific Objectives:

The objective of this course is to provide an introduction to the theory of computation covering the following branches of theoretical computer science:

- To formalize of the notion of problems via formal languages
- To formalize of the notion of computation using "abstract computing devices" called automata
- To understand a hierarchy of classes of problems or formal languages (regular, context-free, context-sensitive, decidable, and undecidable)
- To understand a hierarchy of classes of automata (finite automata, pushdown automata, and Turing machines)

Course Contents:

Basic concepts and definitions; Set operations; partition of a set; Equivalence relations; Properties on relation on set; Proving Equivalences about Sets; Central concepts of Automata Theory: Deterministic Finite Automata (DFA); Minimization of DFA; Non-Deterministic Finite Automata (NDFA); Equivalence of Deterministic and Non-Deterministic Finite Automata; Equivalence between DFA, NFA, NFA; Regular Expressions; Operations on Regular expressions Finite Automata and Regular Expressions; Conversion from FA and regular expressions; Context-Free Grammars; Parse Trees; Ambiguity in Grammars and Languages; Standard Forms; Chomsky Normal Forms; Greibach normal Forms; Minimization of CFG's; Pushdown Automata (PDA); Deterministic and Non-Deterministic (PDA); Formal definition of NPDA; Transition functions of NPDA; NPDA Execution; Accepting Strings with NPDA; Equivalence of PDAs and CFG; The Turing Machine.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1102

References:

1. John Hopcroft, Rajeev Motowani, and Jeffrey Ullman, *Automata Theory, Languages, and Computation*. (3rd Edition)
2. Introduction to Languages and the Theory of Computation - 4th edition John Martin, 2006.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2111	**Computer Networking**	**Theory: 3**	**15 × 3 = 45**

Introduction of the Course:

In this era or Internet, knowledge of computer networking is an invaluable asset for prospective software engineers. This course aims to provide an introduction to fundamental concepts in the design and implementation of computer networks, their protocols, and applications. Major topics to be covered include: computer networking architecture, applications, transport, congestion, routing, and data link protocols, addressing, local area networks and wireless networks. After completing the course, students will be familiar with the fundamental concepts of data communications and computer networking. With the Internet as an example, a student should get an in-depth knowledge on the operations of different protocols of TCP/IP protocol suites and how they interact with each other.

Specific Objectives:

- To learn about the Internet structure
- To know the details of the protocols that are used for successful communication using the Internet.
- To be able to understand how using the Layered approach a packet traverses from source to destination with the help of different protocols used in different layers.

Course Contents: Overview of the Internet, Overview of Networking Protocols, Network Edge, Network Core, Protocol Layers, Internet Structure, Delays; Application Layer: Principles of Networking Applications, Web and HTTP, FTP, SMTP, DNS; Transport Layer: Transport Layer Services, Multiplexing and De multiplexing, Connection-less Transport: UDP, Connection-Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control; Network Layer: Inside a Router, Details of the Internet Protocol (IP), IPv4 Subnetting, SDN Concepts, Routing Algorithms (Link State, Distance Vector), Intra-AS routing in the Internet: Open Shortest Path First (OSPF), Routing among the ISPs: Border Gateway Protocol (BGP); Link Layer: Multiple Access Protocols, Switched LAN, 802.11 Wireless LANs

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1215

References:

1. Computer Networking- A top down Approach -Kurose and Ross
2. Data Communication & Networking – Behrouz Forouzan- McGraw Hill Education
3. Computer Network –Tannenbaum – Pearson Education

Course No.	Course Title	No. of Credits	Credit Hours
MATH 2116	**Linear Algebra**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course:

Ideas of calculus include rates of change of a function which is known as the derivative of a function and integrals which is, in particular, reverse process of derivatives. This course module concentrates on in depth understanding of these two concepts and their properties to communicate the results in applications of Calculus in science, engineering through modeling natural phenomena.

Specific Objective:

- To gain familiarity with key ideas of precalculus, including the manipulation of equations and elementary functions
- To develop fluency with the preliminary methodology of tangents and limits, and the definition of a derivative
- To develop and practice methods of differential calculus with applications
- To develop and practice methods of the integral calculus

Course Contents:

Matrix Algebra, Determinant and its properties, Inverse of a square matrices and properties. Elementary matrices and reduction of a matrix to row echelon form and reduced row echelon forms and their application in solving system of linear equations. Vector space and subspaces of a vector space, Linear span and independence of vectors, basis and dimension of a vector space. Row space, Colum Space, Null space, and Null space of the transpose of a matrix. Orthogonality, Orthogonal projection matrices, Orthonormal bases, Gram-Schmidt orthogonalization algorithm and QR factorization. Eigenvalues, Eigenvectors, Orthonormal diagonalization, Diagonalization of symmetric matrices. Normal and Positive definite matrices. Linear transformation and their matrices, Change of bases, Geometrically important linear transformations. Singular Value Decomposition (SVD), Geometric interpretation of SVD, Principal Component Analysis (PCA) from SVD. Jacobi and Gauss-Seidel methods for approximating solutions of linear system. Power Method for approximating Eigenvalues and Eigenvectors.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

4. Linear Algebra with Applications, 8th Edition, Steven J. Leon.
5. Introduction to Linear Algebra, 5th Edition, Gilbert Strang.

Course No.	Course Title	No. of Credits	Credit Hours
SE 2105	**Software Project Lab I**	**Lab: 3**	15 × (3×2) = 90

Introduction of the Course: Each of the students should complete the software project individually. Students are required to develop sufficiently large software which requires significant "problem solving" effort. Besides, students must showcase the skills they have acquired from their so far completed courses.

Specific Objectives:

- To master coding skills
- To learn to create a complete software
- To learn to manage large codebase
- To keep track of changes to the code through version controlling systems

Prerequisite: SE 1206, CSE 1215, STAT 1203, MATH 1216

Course No.	Course Title	No. of Credits	Credit Hours
BUS 2117	**Business Communication**	**Theory: 3**	**15 × 3 = 45**

Course Contents:

Communication Concept: The Role of Communication in Business, Importance of Communication Skills, Main Form of Business Communication, Process of Human Communication. **Fundamentals of Business Writing:** Adaptation and the Selection of Words, Construction of Clear Sentences and Paragraphs, Writing for Effect. **Basic Pattern of Business Messages:** Directness in Good News and Neutral Situations, Indirectness in Bad Message, Indirectness in Persuasion Message, Letter and Memorandum, Letter Writing Styles, Pattern Variations in Memorandums and the Email, Job Search Activities: Strategies in the Job Search Process, Job search activities, Writing CV, Facing Interviews, Feedback letters for goodwill. **Fundamentals of Report Writing:** Basics of Report Writing, Report Structure: Short Forms, Long and Formal Report, Usages of Graphics. **Other Form of Business Communication:** Informal Oral Communication, Technology-Enabled Communication.**Psychology in Business:**

Job Analysis: Job-oriented Approach, Person-oriented Approach, Purposes of Job Analysis, Methods of Job Analysis, Job Evaluation, Assessment Methods for Selection and Placement: Productive and Counterproductive Employee Behavior: Productive Behavior, Job Performance; Counterproductive Behavior, Withdrawal, Aggression, Mistreatment, Sabotage, and Theft.

Instruction Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: N/A

References:

1. Industrial and Organizational Psychology: Research and Practice, Paul E. Spector, 5th Edition
2. Raymond V. Lesikar, John D. Pettit, Maire E. Flatley, Lesikar's Basic Business Communication, McGraw Hill

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2115L	**Introduction to Algorithm Lab**	**Lab: 1.5**	**15 × (1.5×2) = 45**

Course Contents: Review of Algorithms, Analyzing & Designing Algorithms, Greedy Algorithms - Introduction to Greedy Algorithms, Greedy Choice Property, Dynamic Programming, Fractional Knapsack Problem, Activity Selection Problem, Huffman Encoding, Task Scheduling Problem, Coin Changing Problem, Kruskal's and Prim's Minimum Spanning Tree Algorithms; Divide and Conquer Algorithms - Introduction to Divide and Conquer Design Technique, RunTime Analysis; Dynamic Programming - Introduction to Dynamic Programming Technique, Principle of Optimality, Optimal Substructure Property, Assembly Line Scheduling, Matrix Chain Multiplication, LCS, Viterbi Algorithm, Bitonic Euclidean, Traveling Salesperson Problem; Graph Searching and Shortest Path Problems - Breadth First Search, Depth First Search, Flow Networks, Single Source and All Pair Shortest Path Algorithms; String Matching Algorithms, Approximation Algorithms.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1215, SE 1206

References:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2111L	**Computer Networking Lab**	**Lab: .75**	**15 × (.75×2) = 22.5**

Course Contents: Implementation and relevant lab work of the corresponding theory course.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1215

References:

1. Computer Networking - A top down Approach, Kurose and Ross.
2. Data Communication & Networking, Behrouz Forouzan, McGraw Hill Education.
3. Computer Network, Tannenbaum, Pearson Education.

Semester 4 (2^{nd} year 2^{nd} Semester)

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2201	**Operating System and System Programming**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course:

This course contains different aspects of operating system design and implementation. The operating system provides a convenient and efficient interface between user programs and the bare hardware of the computer on which they run. The operating system is responsible for sharing resources, providing common services needed by many different programs and protecting individual programs from interfering with one another. The course will start with a brief historical perspective of the evolution of operating systems and then cover the major components of most operating systems. This discussion will cover the tradeoffs that can be made between performance and functionality during the design and implementation of an operating system. Particular emphasis will be given to three major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), and file systems; and on operating system support for distributed systems.

Specific Objectives:

- To understand the services provided by and the design of an operating system.
- To understand the structure and organization of the file system.
- To understand what a process is and how processes are synchronized and scheduled.
- To understand different approaches to memory management.
- Students should be able to use system calls for managing processes, memory and the file system.

Course Contents:

Introduction: What is an operating system? History of operating system Operating system concepts Operating system structure

Processes and Threads: Processes state, operations on process, inter-process communications Threads and concurrency Process and thread scheduling

Process synchronization: Synchronizations tools, Principles of Concurrency, Critical Region, Mutual Exclusion, Semaphores and Mutex, Monitors, Classical Problems of Synchronization: Readers-Writers Problem, Producer Consumer problem, dinning philosopher problem.
Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection Recovery from deadlock, An Integrated Deadlock Strategies, Other Issues: Two phase locking, Communication Deadlock, Livelock, Starvation
Memory Management: Memory address, Swapping and Managing, Free Memory Space Virtual Memory Management, Paging Segmentation, Paged Segmentation, Demand Paging Performance, Page Replacement Algorithms, Allocation of Frames, Thrashing
File Systems: Files, Directories, File system management, Input /Output, Principles of I/O, hardware Principles of I/O, software I/O, software layers, Disks.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1211, CSE 2115, SE 2112

References:

1. Silberschatz A.,Galvin P., Gagne G., "Operating System Concepts", 10 th Edition, John Wiley and Sons,
2. Andrew S. Tanenbaum, "Modern Operating Systems", 5th Edition, PHI
3. Stalling William, "Operating Systems", 8 th Edition, Pearson Education
4. Milan Milenkovic, "Operating Systems Concepts and Design", TMGH
5. Das Sumitabha, "Unix Concepts and Applications", 3rd Edition, Tata McGraw Hill, 2003

Course No.	Course Title	No. of Credits	Credit Hours
SE 2215	**Design Patterns**	**Theory: 1.5**	15 × (1.5×1) = 22.5

Introduction of the Course:

This course will illustrate the advanced topics of object oriented concepts, code smells, refactoring and design patterns. The course will guide the students to use these concepts in real life problems. It is expected that, by the end of this course, students will be able to design problems using design patterns and refactor code smells from their programs. In the Lab part of this course, students will implement all the theory concepts using any Object Oriented (OO) programming language.

Specific Objectives:

- To learn the fundamental principles of OO analysis, design, and development
- To know about Code Smells, and to learn to avoid and refactor those
- To learn common Design Patterns in OO design and to implement those appropriately

Course Contents:

Concepts of Object Oriented Design and Design Principles: Revision of OO Concepts, Using UML Diagrams, aggregation, composition and association, Coupling and Cohesion, (SOLID) - Single responsibility principle, Open-Closed Principle, Liskov substitution principle, Interface segregation principle, Dependency Inversion principle. **Refactoring code smells:** Different type of code smells - Inappropriate Naming, Comments, Dead Code, Duplicated code, Primitive Obsession, Large Class, Lazy Class, Alternative Class with Different Interface, Long Method, Long Parameter List, Switch Statements, Speculative Generality, Oddball Solution, Feature Envy, Refused Bequest, Black Sheep and Train Wreck, **Design Patterns:** Importance of learning design patterns, Types of Design Patterns - Structural, Behavioral and Creational Patterns, Creational Patterns – Singleton, Factory, Factory Method, Abstract Factory, Builder, Prototype and Object Pool, Behavioral Patterns - Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, Strategy, Template Method, Visitor and Null Object, Structural Patterns – Adapter, Bridge, Composite, Decorator, Flyweight and Proxy.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: SE 1206.

References:

1. Gamma, Erich. *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1995.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2204	**Database Management System**	**Theory: 3**	15 × (1×3) = 45

Introduction of the Course:

This course will illustrate the database system concepts that include design and development of a database, and its manipulations. The course will guide a student to use the concepts of databases in real life problems. It is expected that, by the end of this course, students will be able to design problems using Entity relation diagram, able to apply normalization if necessary and learn to manipulate data using structured query language and their optimization The course will also guide a student to use the concepts of parallel and distributed database with its recovery.

Specific Objectives:

- To have a broad understanding of database concepts and database management system
- To have a high-level understanding of major DBMS components and their function
- To be able to model an application's data requirements using conceptual modeling tools like ER diagrams and design database schemas based on the conceptual model.
- To be able to apply normalization if necessary
- To be able to efficiently manipulate data using structured query language
- To have understanding about recovery management
- To have understanding about parallel and distributed databases.

Course Contents:

Introduction to Database Systems: Evolution of file processing systems, role of databases in organizations, core components of a database environment; Data Modeling: the Entity-Relationship Diagram; The Relational Model and Normalization: relational model, normalization, transformation of an entity-relationship data diagram into a relational model, express query in relational algebra; Indexing and Hashing: Indexing, B+ tree, static and dynamic hashing; Query Processing and Query Optimization: Measures of Query Cost, Selection Operation, Sorting, Join Operation, Statistical Information for Cost Estimation, Cost-based optimization; Transaction; Concurrency Control: Lock-Based Protocols, Timestamp-Based Protocols; Recovery System: Recovery and Atomicity, Log-Based Recovery; Database System Architectures: Centralized and Client-Server Systems, Server System Architectures; Parallel Databases:I/O Parallelism, Interquery Parallelism. Intraquery Parallelism, Intraoperation Parallelism, Interoperation Parallelism; Distributed Databases: Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control in Distributed Databases.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1215, STAT 1203.

References:

1. Avi Silberschatz, Henry F. Korth, S. Sudarshan; Database System Concepts, 5th Edition
2. Raghu Ramakrishnan, Johannes Gehrke' Database Management Systems, 3rd Edition

Course No.	Course Title	No. of Credits	Credit Hours
BUS 2205	**Business Studies for Engineers**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course: This course introduces the management principles, accounting basics and the basics of financial statements. This course will enable students to take effective decisions in managing the people and process of an organization and to understand the accounting principles.

Specific Objectives:

- To understand the management principles
- To learn the principles and practices of organizational effectiveness and cultures
- To be able to prepare financial statements
- To be able to analyze financial statements and balance sheets

Course Contents: Managers and Entrepreneurs: Management Defined, Role of a Manager, Small-Business Management, The Evolution of Management Thought, Organization, Organization Charts, Contrasting Theories of Organization, Organizational Effectiveness, Organizational Cultures, Change, Conflict, and Negotiation in Organization; The Strategic Management Process, Strategic Implementation and Control, Forecasting. **Accounting Basic:** Forms of Business Organization, Types of Activities performed by Business Organization, Financial statements of Business Organization, The Accounting Equation, The Account and Rules of Debit and Credit, The Journal: Recording of Transaction, Adjusting the Accounts, Closing Entries, and Preparing Financial statements from the Work Sheet. **Analysis and Interpretation of Financial Statement:** Objectives of Financial Statement Analysis, Analysis of a Balance Sheet, Analysis of Statement of Income and Retained Earnings, Ratio Analysis: Liquidity Ratios, Equity or Long Term Solvency Ratio, Profitability Test, Market Test.

Instruction Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: BUS 2117,

References:

1. Stephen P. Robbins and Mary Coulter, *Management*, Prentice Hall, Latest Edition
2. Jerry J. Weygandt, Donald E. Kieso, and Paul D. Kimmel, *Accounting Principles*, Wiley, 8th Ed

Course No.	Course Title	No. of Credits	Credit Hours
SE 2206	**Software Requirements Specification and Analysis**	**Theory: 1.5**	**15 × (1×1.5) = 22.5**

Introduction of the Course:

The purpose of this course is to define the system as a set of assorted ideas. The course will show how to collect and analyze all those ideas that have come up, that is the requirements with respect to consumers. In short, the purpose is to provide a detailed overview of a software product, its parameters and goals. Nonetheless, it also helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

Specific Objectives:

- To understand the software engineering and it's principles
- To apprehend the state of the art and prescribed engineering processes
- To learn the requirements and requirements engineering
- To create requirements models using UML and others
- To visualize the problem in graphical view
- To handle a small to medium scale software project from initiation to modeling
- To be able to document the requirements for the designers and developers

Course Contents:

Review of – The Nature of Software, Software Engineering, The Software Process, Software Engineering Practices, Generic Software Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Model and Agile Development.

Requirements Engineering, Establishing the ground work, Eliciting Requirements, Negotiating Requirements, Validating Requirements, Requirements Analysis, Scenario-Based Modeling, UML Models, Class Based Modeling, Requirements Modeling Strategies, Flow-Oriented Model, Behavioral Model.

Instructional Strategies:

- The medium of instruction is English.
- Lecture materials: recommended books, ppt files, and documents.
- Discuss experimental results to learn analyzing techniques.

Prerequisite: SE 1106, BUS 2117, SE 1206.

References:

1. R. S. Pressman, Software Engineering. A Practitioner's Approach, 7th or higher, McGraw Hill
2. Ian Sommerville. Software Engineering, 9th or higher Edition, Addison-Wesley.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2201L	**Operating System and System Programming Lab**	**Lab: 1.5**	**15 × (1.5×2) = 45**

Course Contents:

Basic of Unix Shell commands, Programs using the system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir; Introduction to shell programs; Process creation and thread creation. Problem solving using process and thread; Implementation of CPU scheduling algorithms, FCFS, SJF, Priority and Round Robin algorithm; Introduction to MUTEX, semaphore and implementation of classical synchronization problems using semaphore; Inter Process Communication using shared memory; Implementation of Banker's algorithm for deadlock detection; Implementation of memory allocation methods using fixed partitioned; Implementation of paging techniques; File & Buffered I/O, directory management in Linux, Basic Bash scripting in Linux, Socket programming in C; Writing a simple device driver.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 2115, SE 2112

References:

1. Linux System Programming, Talking directly from the Kernel and C Library- by Robert Love (2nd edition)

Course No.	Course Title	No. of Credits	Credit Hours
SE 2215L	**Design Patterns Lab**	**Lab: 1.5**	**15 × (1.5×2) = 45**

Introduction of the Course:

This course will illustrate the advanced topics of object oriented concepts, code smells, refactoring and design patterns. The course will guide the students to use these concepts in real life problems. It is expected that, by the end of this course, students will be able to design problems using design patterns and refactor code smells from their programs. In the Lab part of this course, students will implement all the theory concepts using any Object Oriented (OO) programming language.

Specific Objectives:

- To understand the fundamental principles of OO analysis, design, and development
- To Learn Code Smells, and learning to avoid and refactor those
- To learn about the common Design Patterns in OO design and implementing those appropriately

Course Contents:

Implementing a scenario following the SOLID principle; Refactoring the following type of code smells from existing open source projects: Inappropriate Naming, Comments, Dead Code, Duplicated code, Primitive Obsession, Large Class, Lazy Class, Alternative Class with Different Interface, Long Method, Long Parameter List, Switch Statements, Speculative Generality, Oddball Solution, Feature Envy, Refused Bequest, Black Sheep and Train Wreck, Implementation of a scenario for the following design patterns: Creational Patterns – Singleton, Factory, Factory Method, Abstract Factory, Builder, Prototype and Object Pool, Behavioral Patterns - Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, Strategy, Template Method, Visitor and Null Object, Structural Patterns – Adapter, Bridge, Composite, Decorator, Flyweight and Proxy.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, source codes and documents

Prerequisite: SE 1206.

References:

1. Gamma, Erich. Design patterns: elements of reusable object-oriented software. Pearson Education, 1995.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 2204L	**Database Management System Lab**	**Lab: 1.5**	**15 × (1.5×2) = 45**

Course Contents:

SQL: DDL and DML commands; query using select and where clause, rename operation, string operation, set operation, aggregate function, group by, having clause, nested query, in, all, some construct; PL/SQL: named and anonymous function, procedure, trigger, cursor; database administration: Creating Tables, Clusters and Database Tuning; Control File Maintenance; Managing Indexes; Privileges and Roles;

SQL: Creation, modification, configuration, and deletion of databases using SQL, Creation of database schema - DDL (create tables, set constraints, enforce relationships, create indices, delete and modify tables); Database initialization - Data insert, Data import to a database (bulk import using SQL Commands), Practice SQL commands for DML (using select and where clause, rename operation, string operation, set operation, aggregate function, group by, having clause, nested query, in, all, some construct); PL-SQL: use of conditional statement, different types of Loops; use of function, procedure and cursor; database administration: Creating Tables, Clusters and Database Tuning; Control File Maintenance; Managing Indexes; Privileges and Roles;

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1215, STAT 1203.

References:

1. Avi Silberschatz, Henry F. Korth, S. Sudarshan; Database System Concepts, 5th Edition.
2. Raghu Ramakrishnan, Johannes Gehrke' Database Management Systems, 3rd Edition.

Course No.	Course Title	No. of Credits	Credit Hours
SE 2206L	**Software Requirements Specification and Analysis Lab**	**Lab: 1.5**	15× (2×1.5) = **45**

Course Contents:

One small to medium sized real life system will be given to all the students for analyzing in the classroom. Every week students have to bring their own models (Scenario, Data, Object and Behavioral) in the class and all the students will analyze and learn from each other's models.

Two real life mid-scale systems will be distributed among groups (created randomly) of 5/6 students to analyze (one project per group). The output of the analysis will be specification reports.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques

Prerequisite: SE 1106, BUS 2117, SE 1206.

References:

1. R. S. Pressman, Software Engineering. A Practitioner's Approach, 7th Edition or higher, McGraw Hill.
2. Ian Sommerville. Software Engineering, 9th or higher Edition, Addison-Wesley.

Semester 5 (3^{rd} year 1^{st} Semester)

Course No.	Course Title	No. of Credits	Credit Hours
SE 3115	**Software Testing & Quality Assurance**	**Theory: 1.5**	15 × (1.5×1) = 22.5

Introduction of the Course:

Software testing is particularly important in the Software Development Life Cycle (SDLC) since it ensures customer's reliability and satisfaction towards the application. Due to its importance, testing is often considered as a process which is parallel to every software development activity. The testing process can be broadly planned into two activities – Verification and Validation (V&V). A systematic realization of these activities can enhance software quality to a great extent. The aim of this course is to illustrate these aspects of software testing as a methodical process to produce quality software. It is expected that students, at the end of this course, will have a clear understanding of the foundations, methodologies and tools in the area of software testing.

Specific Objectives:

- To learn fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to plan a test project, design test cases and data, conduct testing operations, manage software problems and defects, and generate a testing report.
- To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.To gain the techniques and skills on how to use modern software testing tools to support software testing projects.
- To understand software test automation problems and solutions.
- To learn how to write software testing documents, and communicate with engineers in various forms.

Course Contents:

The Psychology and Economics of Software Testing, Software Testing Life Cycle (STLC), Software Testing Terminology and Methodology, V&V Model, Dynamic Black Box Testing – Boundary Value Analysis, Equivalence Partitioning, State Transition based Testing, Decision Table based Testing, Cause-Effect Graphing based Testing and Error Guessing, Dynamic White Box Testing – Basis Path Testing, Data Flow Testing and Mutation Testing, Inspections, Walkthroughs, Technical Reviews, Unit Testing, Integration Testing, Function Testing, System Testing, Acceptance Testing, Regression Testing, Test Management – Test Organization, Test Plan, Test Design and Specifications, Software Metrics, Software Quality, Quality Control and Quality Assurance, Quality Management and Project Management, Software Quality Metrics, Testing Internet Applications - Security and Performance Testing, Debugging, Test Driven Development (TDD), Behavior Driven Development (BDD).

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: SE 1106, SE 1206, CSE 1215.

References:

1. Naresh Chauhan, Software Testing: Principles and Practices, 1st or higher Edition, Oxford University Press.
2. Glenford J. Myers, Corey Sandler, and Tom Badgett. The Art of Software Testing, 3rd or higher Edition, John Wiley & Sons.
3. Lisa Crispin and Janet Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams, 1st or higher Edition, Pearson Education.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 3102	**Web Technology**	**Theory: 1.5**	**15 × (1.5×1) = 22.5**

Introduction of the Course:

This course is an overview of the modern Web technologies used for Web development. The purpose of this course is to give students the basic understanding of how things work in the Web world from the technology point of view as well as to give the basic overview of the different technologies. The idea of this course is not that the students will learn how to use all of these technologies, but to help them understand the basics and find out where to start.

Specific Objectives:

- To be able to get an introduction about various Scripting Languages.
- To be able to provide an up-to-date survey of developments in Web Technologies.
- To be able to know techniques involved to support real-time Software development.
- To be able to successfully perform and interact in a technology-driven society.
- To describe and explain the relationship among HTML, XHTML, CSS, JavaScript, XML, ASP.NET and other Web technologies
- To get familiar with W3C standards and recommendations.
- To create and publish a basic web page using HTML and its many tags;
- To describe limitations of creating interactivity including browser support and differences;
- To describe the difference between Java and JavaScript;
- To understand and use JavaScript variables, control structures, functions, arrays, and objects;
- learn and modify CSS properties using JavaScript;
- Create server side applications using ASP.NET.

Course Contents:

Introduction to Html, Java Script & CSS, Server Side Programming: HTTP Server, Application Server, MVC Web Framework, Web Services, Database Access: Object Relational Mapping, Lambda Expression, Language Integrated Query, Data Reader, Writer, Web Security: Denial of Service, Buffer Overflow, Cross Site Scripting, Authentication and Access Control.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: SE 2215, CSE 2204, CSE 2111.

References:

1. Deitel & Deitel, Goldberg, "Internet and world wide web – How to Program", Pearson Education Asia, 2001.

2. Rajkamal, "Web Technology", Tata McGraw-Hill, 2001.

3. Teach yourself web technologies part I & II- I. Bayross, BPB

4. Web Design in a Nutshell- J. Niederst, SPD

Course No.	Course Title	No. of Credits	Credit Hours
CSE 3116	**Mathematical and Statistical Methods for Data Analysis**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course:

This course will help to learn the usages of Mathematics and Statistics for solving different real life problems. This course is designed to provide you with a comprehensive understanding of the mathematical and statistical methods that are essential for data analysis. Through a rigorous curriculum that combines theoretical concepts, practical applications, and hands-on exercises, students will develop the skills necessary to analyze complex data sets, draw meaningful insights, and make informed decisions based on quantitative evidence.

Specific Objectives:

- Understand the probabilistic and statistical foundations of data analytics
- Understand and apply linear algebra for data analytics.

Course Contents:

Probability and distribution: . Hypothesis Testing: ANOVA, . Stochastic Process: Markov chains, HMM. Matrix Factorization and Applications: QR factorization, LU decomposition, Singular value decomposition (SVD), Dimension reduction via PCA, SVD, graph partitioning, introduction to spectral graph theory, k-means clustering and its relation to matrix factorization. Advanced Regression Analysis: Linear least Squares, . Optimization: Normal equations and gradient descent. Information Theory: mutual information, entropy.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: STAT 1203, MATH 1216, MATH 2116, CSE 2115.

References:

1. Matrix Computations, 3rd Edition, Gene H. Golub, Charles F. Van Loan.
2. Linear Algebra with Applications, 8th Edition, Steven J. Leon.
3. Introduction to Linear Algebra, 5th Edition, Gilbert Strang.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 3117	**Artificial Intelligence**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course:

This is an introductory course to provide the fundamental knowledge of artificial intelligence. The course discusses search algorithms, probability as a mathematical tool, Bayesian networks, logic and planning algorithms, machine learning algorithms including neural networks. Application domains such as robotics and natural language processing. The objective is to give an overall idea about the field without delving into advanced details.

Specific Objectives:

- To introduce the functions of modern Artificial Intelligence.
- To explain how Artificial Intelligence enables capabilities that are beyond conventional technology, for example, chess-playing computers, self-driving cars, robotic vacuum cleaners.
- To understand and implement search and adversarial (game) algorithms.
- To understand mathematical models such as Bayesian networks and apply them to a range of AI problems.
- To introduce different logic formalisms and decision taking in planning problems.
- To give an overall understanding of machine learning algorithms and extracting knowledge models from data.
- To clarify the application of AI in Natural Language Processing and Robotics.

Course Contents:

The Foundations and history of Artificial Intelligence, The State of the Art Agents and Environments, The Concept of Rationality,Problem-Solving Agents, Uninformed Search, Informed (Heuristic) Search, State-of-the-Art Game Programs,Constraint Satisfaction ProblemsBayes' Rule and Its Use, Probabilistic Reasoning, Bayesian Networks, Inference in Temporal Models, Hidden Markov Models, Dynamic Bayesian Networks,Propositional Logic, First-Order Logic, Knowledge Representation, Ontological Engineering, Supervised and unsupervised Learning, Learning Decision Trees, Artificial Neural Networks, Support Vector

Machines, Ensemble Learning, Brief introduction to Language Models, Text Classification, Information Retrieval, Information Extraction, Syntactic Analysis (Parsing), Machine Translation, Speech Recognition, Perception, Robotic Perception, Moving.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: MATH 1216, STAT 1203, CSE 2115, MATH 2116.

References:

1. Russell, S., & Norvig, P. Artificial intelligence: a modern approach. Third Edition. Pearson new international edition. 2014.
2. Cawsey, A. (1998). The essence of artificial intelligence. Prentice Hall.
3. Poole, D. L. & Mackworth, A. K. (2010). Artificial intelligence: foundations of computational agents. Cambridge University Press.

Course No.	Course Title	No. of Credits	Credit Hours
SE 3118	**Software Design and Analysis**	**Theory: 1.5**	**15 × (1.5×1) = 22.5**

Introduction of the Course:

Software Design is a collection of principles, practices and concepts to produce high-quality software products. The design process of a software system yields specification of the architecture, procedural description of components and user interface design. This course provides in-depth coverage of the concepts needed to effectively design high quality systems combining architectural, component and user interface design.

Specific Objectives:

- After attending this course, students will have a better understanding of:
- To understand the essential considerations in architectural design process
- To learn the systematic process to design an architecture
- To understand the role of architecture evaluation
- To be able to create component level design of software systems
- To understand the role of component-based software development
- To learn the process to design user interface of software systems
- To be able to implement the guidelines to improve the user interface and user experience of software systems

Course Contents: Theory:

Design Concept - The Design Process, Design Concepts, The Design Model; Architectural Design: Software Architecture, Architectural Genres, Architectural Styles, Architectural Design Process, Assessing Alternative Architectural Designs, Architecture and Agility; Component-Level Design: Definition of a Component, Designing Class-Based Components, Conducting Component-Level Design, Component-Level Design for WebApps, Designing Traditional Components, Component-Based Development; User Interface (UI)/ User Experience (UX) Design: The Golden Rules, Nielsen's Heuristics, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Web App Interface Design, Design Evaluation.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: SE 2206, SE 2215.

References:

1. Software Engineering: A Practitioner's Approach, 8th Edition, Roger S. Pressman.
2. Software Engineering, 10th Edition, Ian Sommerville.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 3119	**Cryptography and Security Mechanism**	**Theory: 3**	15 × (1×3) = 45

Introduction of the Course:

This course is an introduction to the broad field of computer, network, and data security. Upon completion of this course, the students

- should be able to explain concepts related to applied cryptography.
- should know the algorithms of symmetric cryptography, asymmetric cryptography, and digital signatures.
- should explain the theory behind the security of different cryptographic algorithms.
- should explain common network vulnerabilities and attacks, defense mechanisms against network attacks, and cryptographic protection mechanisms.

Specific Objectives:

- To acknowledge the students about the fundamentals of cryptography and how cryptography serves as the central language of information security.
- To understand how the knowledge of information security can counteract attempts to the attacks of valuable information technology assets
- To understand the basic software tools for assessing the security posture of a computer or a network

Course Contents:

Introduction to Information Security: Security attacks [active, passive, brute force, cryptanalysis, insider], security services [CIA, AAA], security terminologies [adversary, vulnerability, threat, attack], assumption and trust, security policy and mechanism, threat analysis, attacker modeling, **Introduction to Cryptography:** Plaintext, ciphertext, encryption, decryption, symmetric and asymmetric encryptions, block cipher and stream ciphers, historical cipher techniques. **Modern Cryptography:** DES, need for 3DES, computationally secure cipher, unconditionally secure cipher, types of attack on cryptographic text, cipher mode of operation, Cryptographic algorithm: Detail operation of AES, RC4 Algorithm, **Cryptographic algorithm:** Introduction to modular arithmetic, detail operation of RSA, correctness proof of RSA, **Hash:** Properties of Hash functions, hash algorithm variants, SHA 512, Introduction to HMAC and CMAC, Authentication Mechanism, Password-based authentication, Biometric authentication, Challenge-Response authentication, Multifactor authentication, Access Control and Different Access Control Models, Diffie Hellman (DH) key exchange protocol, Problems in DH key exchange protocol, Public-Key Infrastructure and Digital Signature (Generation and Verification), Risk Management Framework.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1101, CSE 1102, CSE 2111.

References:

1. Cryptography and Network Security- Principles and Practice, William Stallings
2. Computer Security- Principles and Practice, William Stallings and Lawrie Brown
3. Introduction to Computer Security, Matt Bishop
4. Security in Computing, Charles P. Pfleeger, Shari Lawrence Pfleeger and Jonathan Margulies
5. Computer Security: Art and Science, Matt Bishop
6. Network Security: Private Communication in a Public World, Charlie Kaufman, Radia Periman, and Mike Speciner.

Course No.	Course Title	No. of Credits	Credit Hours
SE 3105	**Software Project Lab II**	**Lab: 3**	**15 × (2×3) = 90**

Introduction of the Course: Software engineering ensures quality by following rigorous processes throughout the Software Development Life Cycle (SDLC). Requirement engineering, or the engineered process of collecting, analyzing, specifying and validating requirements is considered the most crucial of all these processes as it determines the functionalities of the software. The goal of SE 3105: Software Project Lab II (SPL II) is to ensure that students follow and apply the processes of requirement engineering while creating software. Therefore, it is necessary that students choose and propose a project that will allow them to apply requirement engineering within the time frame of the project. Moreover, it is also encouraged and suggested that students choose projects with ample scope to gather requirements from stakeholders outside their comfort zone, i.e. from IIT, relatives, and/or friends. Students are required to create a team of 2 members. However, the number of team members can be varied for special cases which will be decided by the assigned course manager.

Specific Objectives:

- To acquire the ability to collect requirements for real-world projects
- To learn to work collaboratively
- To to able to manipulate data intensive projects

Prerequisite: SE 2206, SE 2215.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 3102L	**Web Technology Lab**	**Lab: 1.5**	**15 × (1.5×2) = 45**

Course Contents:

HTML: Basic tags of HTML, Forms and Controls, CSS: Different types of CSS tags and their values, Java Script: Elements of Java Script, different types of objects in JS, DOM manipulation, event handling, Server Side Programming: Introduction to ASP.NET and Visual Studio IDE, Web development using ASP.NET MVC, Database Programming usingEntity Framework with Lambda Expression and Language Integrated Query (LINQ), Web Services.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab re

Prerequisite: SE 2215, CSE 2111.

References:

1. Deitel & Deitel, Goldberg, "Internet and world wide web – How to Program", Pearson Education Asia, 2001.
2. Rajkamal, "Web Technology", Tata McGraw-Hill, 2001.
3. Teach yourself web technologies part I & II- I. Bayross. BPB
4. Web Design in a Nutshell- J. Niederst, SPD

Course No.	Course Title	No. of Credits	Credit Hours
CSE 3119L	**Cryptography and Security Mechanisms Lab**	**Lab: 0.75**	**15 × (0.75×2) = 22.5**

Course Contents: Implementation and relevant lab work of the corresponding theory course.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 1101, CSE 1102, CSE 2111.

References:

1. Cryptography and Network Security- Principles and Practice, 5th Edition, William Stallings
2. Computer Security- Principles and Practice, 2nd Edition, William Stallings and Lawrie Brown
3. Security in Computing - 2nd Edition, Charles P. Pfleeger and Shari Lawrence Pfleeger
4. Introduction to Computer Security- 1st Edition, Matt Bishop
5. Computer Security: Art and Science- 1st Matt Bishop
6. Network Security: Private Communication in a Public World- 2nd Edition- Charlie Kaufman, Radia Periman, and Mike Speciner.
7. Computer Networking- A top down Approach (6th Edition) -Kurose and Ross

Semester 6 (3rd year 2nd Semester)

Course No.	Course Title	No. of Credits	Credit Hours
SE 3215	**Internship**	**Lab: 15**	15 × (2×15) = 450

Introduction of the Course: To gain practical experience, students of the sixth semester will be assigned to a professional establishment as interns. Students, upon completion of 5th semester, will have an acceptable level of theoretical knowledge, which they will be able to apply in practice. At the same time, they will become aware of the practical trends and professional environment. After completion of the internship, students will have to present a report on the respective domain of work.

Specific Objectives:

- To learn to work in a professional environment
- To improve the communication, team-work and other relevant soft skills
- To apply theoretical knowledge to real-life projects

Semester 7 (4^{th} year 1^{st} Semester)

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4115	**Distributed Systems**	**Theory: 1.5**	**15 × (1.5×1) = 22.5**

Introduction of the Course:

This course explains the principles of distributed systems such as communication, naming, synchronization, replication and fault tolerance using examples and case studies. It covers architectures in distributed systems, reflecting the progress that has been made on organizing distributed systems, and new topics such as peer-to-peer computing, sensor networks, web services, grid computing, virtualization, cloud computing and its roots in distributed systems mechanisms, and self-management of distributed systems.

Specific Objectives:

- To understand the principles of distributed systems and apply them
- To understand the architecture of distributed systems
- To understand and apply distributed algorithms
- To design and implement simple distributed systems
- To understand how modern distributed systems are designed

Course Contents:

Foundations: Characterization of DS, System Models, Networking and Inter-networking, Inter-process Communication, Remote Invocation, Indirect Communication and Operating System Support; Middle-ware: Distributed Objects and Components, Web Services and Peer-to-Peer Systems services, Security, Distributed File Systems and Name Services; Distributed algorithms: Time and Global States, Coordination and Agreement Shared data, Transactions and Concurrency Control, Distributed Transactions, and Replication, New challenges: Mobile and Ubiquitous Computing.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite: CSE 2111, CSE 2115.

References:

1. Distributed Systems: Principles and Paradigms , Tanenbum.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4116	**Compiler Design**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course:

The purpose of this course is to learn how to transform programs written in common user-level programming languages into code that can be executed by the processor. This course will cover all the stages of a compiler, from processing program text into an internal representation, through optimization and program improvement, to the generation of assembly code.

Specific Objectives:

On completion of the course, the student should be able to:

- structure a compiler as a sequence of distinct translation steps
- use regular languages to describe the lexical elements of a programming language
- describe lexical analysis using a finite automaton
- use context free languages to describe the syntactic structure of a programming language
- use the parsing methods top-down (recursive descent) and bottom-up (LR)
- use abstract syntax trees to represent the results of the syntactic analysis
- break down statements and expressions to simpler designs, and translate syntax trees to intermediate code

- describe how recursive procedure calls can be implemented by means of stacks, activation posts and machine registers
- translate the simplified intermediate code of a program to machine-specific instructions

Course Contents:

Introduction to Compilers

Structure of a compiler – Lexical Analysis – Role of Lexical Analyzer – Input Buffering –Specification of Tokens – Recognition of Tokens – Lex – Finite Automata – Regular Expressions to Automata – Minimizing DFA.

Syntax Analysis

Role of Parser – Grammars – Error Handling – Context-free grammars – Writing a grammar –Top Down Parsing – General Strategies Recursive Descent Parser Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR (0)Item Construction of SLR Parsing Table -Introduction to LALR Parser – Error Handling and Recovery in Syntax Analyzer-YACC.

Intermediate Code Generation

Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking.

Run-Time Environment and Code Generation

Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management – Issues in Code Generation – Design of a simple Code Generator.

Code Optimization

Principal Sources of Optimization – Peep-hole optimization – DAG- Optimization of Basic Blocks, Global Data Flow Analysis – Efficient Data Flow Algorithm.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: SE 2112, CSE 2115

References:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman.
2. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
3. Compiler Construction, Louden, Thomson.

Course No.	Course Title	No. of Credits	Credit Hours
SE 4117	**IT Laws & Ethics**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course: The course will introduce the different legal mechanisms that software engineers need to know to face different scenarios in their career, and to explain the common cybercrimes and the relevant laws to address those crimes. The course also helps to analyze the different problem scenarios and to find the way-out effectively.

Specific Objectives: Specific Objectives are as follows.

- Understand the legal controls relevant to the protection of ICT-based system clearly
- Explain the various cyber threat/ cybercrime and relevant laws to address the cybercrime
- Examine various problem scenarios to take the right action
- Contribute in national policy development

Course Contents:

Introduction: Introduction to legal aspects, Jurisdiction, Intellectual Property Rights (Copyright and infringement, fair use Doctrine, Software copyright protection, Patent, software patent, software cross-licensing agreements, and Trademark), Contracts and licenses, E-contract, Privacy in the workplace, Trade secrets and non-disclosure agreement; **Computer and Internet Crime:** Threats to information resources, including military and economic espionage, Malware, **IT security incidents:** communications eavesdropping, computer break-ins, denial-of-service, destruction and modification of data, distortion and fabrication of information, forgery, etc. **Relevant Laws:** Cyber laws and rights in today's digital age, Relevant laws and policy guidelines (ICT Act 2006, Digital Security Act 2018, Cyber security strategy guide, Information Security policy for Bank and non-bank financial institutes), Laws relevant to Intellectual Property Rights and E-contract. **Introduction to Ethics:** Definition of Ethics, Moral code, morality, importance of integrity, vices, virtues, Difference between morals, ethics, and law, Fostering good business ethics, establishing corporate code of ethics, Knowledge about ethical work environment. **Ethics for IT Workers and IT Users:** IT Profession, Professional Relationships that IT workers must manage, Ethical issues aroused in IT organizations, Professional code of ethics, professional organizations, certifications, government licensing, issues associated with the government licensing IT Workers, IT Professional malpractices, Common Ethical issues for IT Users, supporting the ethical practice of IT Users. **Freedom of Expression:** Freedom expression, obscene speech, defamation, hate speech, key issues related with freedom of expressions, internet filtering, and anonymous communication. **Ethics in**

software development and for IT organizations: Social networking, Key ethical issues in social networking, digital divide, how to deal with a particular case of any ethical issues in a company. **E-Waste management.**

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and online documents

Prerequisite: SE 1106, CSE 3119.

References:

1. George W. Reynolds, Ethics in Information Technology.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4118	**Machine Learning**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course:

This course offers an introduction to the fundamental concepts in machine learning and useful machine learning algorithms. The basic clustering and dimensionality reduction algorithms will also be covered. Various issues related to the application of machine learning algorithms along with overfitting, bias and variance trade-offs, evaluation strategies and cross-validation will also be discussed.

Specific Objectives:

- To introduce the basic concepts of machine learning.
- To learn the commonly used techniques of machine learning.
- To gain experience of solving problems using machine learning.
- To understand the strengths and weaknesses of different machine learning algorithms.

Course Contents:

Basic definitions, types of learning, evaluation, cross-validation, Linear and logistic Regression, Support Vector Machine, Decision trees and ensemble learning, Instance based learning, feature reduction, probability and bayes learning, Bayesian networks, Dimensionality Reduction, Clustering: k-means, adaptive hierarchical clustering, gaussian mixture model, Neural Networks: Perceptron, multilayer network, backpropagation, Introduction to deep neural networks.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, presentation slides, and notes
- Discuss experimental results to learn analyzing techniques

Prerequisite: MATH 1216, STAT 1203, CSE 2115, MATH 2116.

Reference Books:

1. The Elements of Statistical Learning: Data Mining, Inference, and Prediction by Trevor Hastie, Robert Tibshirani, and Jerome Friedman.
2. Pattern Recognition and Machine Learning by Christopher M. Bishop.
3. Data Mining: Practical Machine Learning Tools and Techniques by Ian H. Witten, Eibe Frank, and Mark A. Hall.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4115L	**Distributed Systems Lab**	**Lab: 1.5**	**15 × (1.5×2) = 45**

Course Contents:

Foundations - Characterization of DS, System Models, Networking and Inter-networking, Inter-process Communication, Remote Invocation, Indirect Communication and Operating System Support Middle-ware - Dist. Objects and Components, Web Services and Peer- to-Peer Systems System services – Security, Distributed File Systems and Name Services Distributed algorithms - Time and Global States, Coordination and Agreement Shared data, Transactions and Concurrency Control, Distributed Transactions, and Replication, New challenges - Mobile and Ubiquitous Computing

Instruction Strategies

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents
- Discuss experimental results to learn analyzing techniques (using lab results)

Prerequisite Courses: CSE 2111, CSE 2115

References:

1. Distributed Systems: Principles and Paradigms, Tanenbum.

Course No.	Course Title	No. of Credits	Credit Hours
SE 4119	**Project**	**Lab: 3**	**15 × (2×3) = 90**

Introduction of the Course: In this course, the students are required to implement a complete software individually by following all the steps of software development life cycle – requirement analysis, design, implementation, testing, deployment.

Specific Objectives:

- To apply industrial experience gathered from internship
- To apply all the skills accumulated throughout their honors life to full-fledged software
- To demonstrate the ability to follow software development life cycle

Semester 8 (4th year 2nd Semester)

Course No.	Course Title	No. of Credits	Credit Hours
SE 4215	**Software Security**	**Theory: 1.5**	15 × (1×1.5) = 22.5

Introduction of the Course:

This course is an introduction to the field of software security. Security has always been left behind during the software design and development phase. After completing this course, students should be able to incorporate security touch points in software development and also, gain practical knowledge about known security attacks and their prevention.

Specific Objectives:

- Learn how to incorporate security touchpoints in software developments
- Gain both theoretical and practical knowledge about

Course Contents: Fundamentals of Computer Security, CIA Principles, Security Services and Mechanisms: User Authentication: Password-based authentication, Token-based authentication, Bio-metric authentication, Remote user authentication; Access Control: Discretionary and Role-based access control, Database Security, Sql Injection Attacks, Buffer Overflow attack, Software security touch points: Code Review, Risk Analysis, Penetration Testing, Abuse Cases, Web-vulnerabilities: Cross-site scripting (XSS), cross-site request forgery (CSRF) attacks, Software-penetration testing, Static code analysis, Coding Errors

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: CSE 3119, CSE 3102, SE 3115.

References;

1. Computer Security: Principles and Practice, by William Stallings and Larry Brown.
2. Software Security: Building Security IN, Gary McGraw.

Course No.	Course Title	No. of Credits	Credit Hours
SE 4203	**Software Project Management**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course: Students who already understand the fundamentals of software engineering and programming and are capable of creating software should enroll in this course. Understanding the fundamental concepts of software project management and the responsibilities of a good project manager are the main objectives of this course. These concepts will be applied and extended with a focus on project management in order to maximize the benefits from each stage of the software project lifecycle.

Specific Objectives:

- To understand basic concepts of software project management
- Be familiar with the different methods and techniques used for project management
- Work in teams to develop a project plan for a project scenario that contains key tasks, a critical path, dependencies, and a realistic timeframe using project management approaches for IT projects.

Course Contents:

Introduction to Project Management (PM): Project Management (PM) fundamentals: The PM field and job market – People, Process, Product, Technology, historical background and evolution. Basic PM skills, SPM framework, elements, stakeholders, boundaries, challenges of SPM. **Software Project Planning:** planning objective, project plan, variations, structure of SPM plan, project estimation, estimation methods, models, and decision process. **PM Organization and Scheduling:** WBS, types of WBS, functions, activities, tasks, life cycles, phasing and purpose of phasing, building project schedule, network diagrams: PERT, CPM, Bar charts, Gantt charts, Risk and Change Management. **Software Project Management Techniques:** Use of methodologies, managing risks and issues, managing quality, configuration, change, crisis, documentation, release. **Project Monitoring and Control:** Dimensions of monitoring and control, earned value indicators (BCWS, CV, SV, CPI, SPI), backlog management, dispute and error tracking, RMMM charts. **System Test Process:** Test specifications, black box and white box testing, Test scripts, unit and integration testing, acceptance test specifications, test tools. **Industry Scenarios:** Domain analysis, business case analysis, dynamicity, success and failure factors, case studies

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, papers, presentation files, and web resources

Prerequisite: SE 1106, CSE 1101, CSE 1215, SE 2105, CSE 2204, SE 3105, SE 518.

References:

1. Stellman, Andrew, and Jennifer Greene. *Applied software project management.* " O'Reilly Media, Inc.", 2005
2. Phillips, Joseph. *IT project management: on track from start to finish.* McGraw-Hill, Inc., 2002.
3. Rubin, Kenneth S. *Essential Scrum: A practical guide to the most popular Agile process.* Addison-Wesley, 2012.
4. Software Project Management (5th ed.), Bob Hughes and Mike Cotterell, McGraw Hill, (2009)

Course No.	Course Title	No. of Credits	Credit Hours
SE 4216	**Software Metrics**	**Theory: 1.5**	**15 × (1×1.5) = 22.5**

Introduction of the Course:

The role of measurement is to assess situations, track progress, evaluate effectiveness, and help determine possible courses of action. Thus, Software Measurement becomes an essential component of good Software Engineering. To do so, an understanding of fundamental measurement and experimentation is necessary. In this course, students will gain ideas of measurement theories, related fundamentals, and specific software engineering-related metrics.

Specific Objectives:

- Understanding the importance and role of software metrics in software engineering
- To apply software metrics to software
- Understanding the tradeoffs associated with different types of software metrics
- Applying software metrics to properly measure different aspects of software engineering processes

Course Contents:

Overview and scopes of Software Metrics: measurement basics, understanding, control, and improvement, **The Basics of Measurement**: representation theory, measurement scales and scale types, meaningfulness in measurement, **Goal-based framework for software measurement**: determining what to measure, applying the different framework, validation, **Empirical Investigation**: control variables and study type, study goals and hypothesis testing, threats to validity, planning case studies, **Data collection and reliability measurement:** Classical data analysis techniques, hypothesis testing, sample analysis techniques, an overview of statistical tests, **Measuring Internal Attributes: Size and Structure**: code size, design size, functional size, application of size measurements, **Measuring Cost and Effort**: Cocomo models, function point analysis, **Measuring External Attributes: Quality and Reliability**: modeling software quality, usability measures, maintainability measures, and security measures. Basic reliability theories, parametric reliability growth models.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, papers, presentation files, and web resources

Prerequisite: SE 3115, CSE 3116, SE 3118.

References:

1. Software Metrics, Norman Fenton and James Bieman.

Course No.	Course Title	No. of Credits	Credit Hours
SE 4201	**Project**	**Lab: 3**	15 × (2×3) = 90

Introduction of the Course: In this course, the students are required to implement a complete software individually by following all the steps of the software development life cycle – requirement analysis, design, implementation, testing, deployment.

Specific Objectives:

- To apply industrial experience gathered from internship
- To apply all the skills accumulated throughout their honors life to full-fledged software
- To demonstrate the ability to follow software development life cycle

Prerequisite: N/A

Course No.	Course Title	No. of Credits	Credit Hours
SE 4215L	**Software Security Lab**	**Lab: 1.5**	15 × (1.5×2) = 45

Course Contents:

Demonstration Buffer Overflow, XSS, CSRF, SQL Injections attacks, Static code analysis.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, papers, presentation files, and web resources
-

Prerequisite: CSE 3119, CSE 3102, SE 3115.

References:

1. Computer Security: Principles and Practice, by William Stallings and Larry Brown.
2. Software Security: Building Security IN, Gary McGraw.

Course No.	Course Title	No. of Credits	Credit Hours
SE 4216	**Software Metrics**	**Theory: 1.5**	**15 × (2×0.75) = 22.5**

Course Contents: Implementation and relevant lab work of the corresponding theory course.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite: SE 3115, CSE 3116, SE 3118.

References:

1. Software Metrics, Norman Fenton and James Bieman.

Elective Courses

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4270	**Image Processing**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course:

The objective of this course is to introduce the basic ideas behind digital image processing. The topics that will be studied in this course include visual perception, image representation, image enhancement in the spatial and frequency domains, image restoration, color image processing, image compression, feature extraction for pattern recognition, and classification and segmentation in supervised and unsupervised techniques. The student will have a basic understanding of image processing at the end of this course and be able to grasp how digital images are processed.

Specific Objective:

- To gain the theoretical and practical knowledge of basic algorithms of image processing and pattern recognition
- To identify, formulate, and solve problems related to the recognition of objects in an image.
- To select a suitable classification process, features, and appropriate classifier to address and solve a desired pattern recognition problem.

Course Contents: Digital Image Fundamentals - Elements of Visual Perception, Light and the Electromagnetic Spectrum, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships between Pixels, Linear and Nonlinear Operations; Image Enhancement in the Spatial Domain - Background, Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Combining Spatial Enhancement Methods; Image Enhancement in the Frequency Domain – Background, Introduction to the Fourier Transform and the Frequency Domain, Smoothing Filters, Sharpening Filters; Image Restoration - A Model of the Image Degradation/Restoration Process, Noise Models, Estimating the Degradation Function, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Constrained Least Squares Filtering, Geometric Mean Filter, Geometric Transformations; Image Segmentation- Point, Line, and Edge Detection, K-mean Clustering, Canny Edge Detection, Otsu method; Color Image Processing - Color Fundamentals. Color Models, Pseudo Color Image Processing, Color Transformations, Color Segmentation, Noise in Color Images; Image Compression - Fundamentals. Image Compression Models, Error-Free Compression, Lossy Compression, Image Compression Standards; Morphological Image Processing – Preliminaries, Dilation and Erosion, Opening and Closing, The Hit-or-Miss Transformation, Basic Morphological Algorithms; Image Segmentation - Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation, Boundary Descriptors, Regional Descriptors; Pattern Recognition - *Fundamentals of Pattern Recognition, Local and Global* features, *Hough* Transformation, Local Binary Pattern, HoG features, Statistical Pattern Recognition, Principal Component Analysis; Classification and Segmentation- Fundamentals of Classification and Segmentation, Support Vector Machine, Random Forest; Hyperspectral Image Fundamentals; 3D Point Cloud Fundamentals.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, research papers, and documents

Prerequisite:

References:

1. Digital Image Processing - Rafael C Gonzalez and Richard E. Woods.
2. Pattern Recognition- William Gibson
3. Pattern Classification – R.O. Duda, P.E. Hart, and D.G. Stork

Course No.	Course Title	No. of Credits	Credit Hours
SE 4290	**Software Maintenance**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course: This course covers the last step of software development life cycle. Through this course, students learn how to keep a system operational by fixing bug, recovering from failure, and accommodating changes.

Specific Objective:

- To comprehend the fundamental concepts of software maintenance
- To understand the existing software systems
- To add or modify an existing software system
- To determine the impact of software changes
- To improve the quality of a software

Course Contents:

Software Maintenance Fundamentals: what is software maintenance, why is it important, types of software maintenance; Software Maintenance Measurement: McCabe's Cyclomatic Complexity, Halstead Metrics; Maintenance Process: Maintenance Models, IEEE/EIA 1219 Standard for Maintenance Process, Program Comprehension: Cognition Models for Program Comprehension, Understanding Factors that Affect Program Comprehension, Reengineering: General Model For Software Reengineering, Software Reengineering Strategies, Reverse Engineering: Techniques Used for Reverse Engineering, Change Impact Analysis, Refactoring, DevOps: Towards Easing Software Maintenance: Git Branching Strategies, Observability.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, research papers and documents

Prerequisite: N/A

References:

1. Software Maintenance: Concepts and Practice By Penny Grubb, Armstrong A. Takang 2nd edition World Scientific USA.
2. Software Evolution and Maintenance: A Practitioner's Approach by Priyadarshi Tripathy, Kshirasagar Naik (2014, Wiley).

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4250	**Natural Language Processing**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course:

Natural Language Processing (NLP) is the study of how computers and human languages interact, with a focus on how to design computers to process and analyze enormous amounts of natural language data. It is a multidisciplinary branch of linguistics, computer science, and artificial intelligence. The ultimate goal of NLP is to create a machine that is able to "understand" the contents of documents, including the subtleties of language used in different contexts. Once the information and insights are accurately extracted from the documents, the technology can classify and arrange the documents. Major applications of NLP include text-to-speech conversion, speech-to-text conversion, language understanding, language generation, machine translation, text summarization, question answering, sentiment analysis, etc.

Specific Objective:

- Understand the basic concepts of natural language and linguistics
- Apply text preprocessing techniques
- Understand and apply vector semantics and word embedding techniques
- Apply machine learning and deep learning models for NLP tasks
- Learn about the applications of NLP

Course Contents:

Basic NLP concepts: Regular expressions, tokenization, lemmatization, stemming, edit distance, N-gram language models, parts of speech, named entity; **Review of classification algorithms:** Naive Bayes, Logistic Regression, Neural networks; **Deep neural networks:** CNN, RNN, LSTM; **Vector semantics and embeddings:** Words and vectors, cosine similarity, TF-IDF, word2vec, skip-gram, fastText; **NLP applications:** language generation, machine translation, text summarization, question answering, sentiment analysis.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, papers, presentation files, and web resources

Prerequisite: CSE 4118

References:

1. Speech and Language Processing, by Daniel Jurafsky and James H. Martin
2. Natural Language Processing with Python, by Steven Bird and Ewan Klein

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4251	**Introduction to Data Mining**	**Theory: 3**	**15 × (3×1) = 45**

Introduction of the Course:

This course is an introductory course on data mining. It introduces the basic concepts, principles, methods, implementation techniques, and applications of data mining. Techniques to find patterns in databases, prediction and forecasting, using different statistical and machine learning techniques, and visualization of the generated structures, data warehousing and on-line analytical processing (OLAP) will also be discussed.

Specific Objective:

- To introduce students to the basic concepts and techniques of Data Mining.
- To develop skills of using data mining tools for solving real world practical problems.
- To provide experience of doing independent analysis on the different Data Mining techniques.

Course Contents:

Introduction to Data mining, Data Warehousing, On-line Analytical Processing; Overview of Clustering, Classification, Rule learning; Data mining process: Data preparation/cleansing and next steps, Association Rule mining, different algorithm types; Classification/Prediction: basics, tree-based approaches, Neural Networks; Clustering: basics, statistical approaches, Neural-net and other approaches; Text Mining, Time Series Mining, Mining Data Streams, Multi-Relational Data Mining, Data Mining for Fraud Detection

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, presentation slides, and notes
- Discuss experimental results to learn analyzing techniques

Prerequisite: CSE517

References:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann, 2005, ISBN: 0-12-088407-0.
2. Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) 3rd Edition

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4271	**Photogrammetry**	**Theory: 3**	15 × (3×1) = 45

Introduction of the Course:

This course provides a thorough understanding of photogrammetry's theoretical and practical ideas. Students will understand perspective geometry using the foundations of optical mathematics. The techniques to extract data from aerial and satellite imaging systems will be taught to the students. This course also covers the processes involved in reconstructing the three-dimensional model of the real environment as well as the elements that affect the production of photos. They will also learn about the techniques and uses of laser scanning. Students who successfully complete this course will be able to organize photogrammetric projects, capture and process images, and deliver fundamental photogrammetric products.

Specific Objective:

- To gain knowledge of the fundamental ideas of image geometry and the measurement of an aerial photograph.
- To get acquainted with 2D and 3D image interpretation and information extraction from ground-based, aerial, and satellite remote sensing and photogrammetric data
- To implement a range of photogrammetric measurement techniques and to apply the theory of photogrammetry to a range of 2D/3D measurement problems.
- To Understand the process of reconstructing a three-dimensional model for the real world

Course Contents:

Introduction to Photogrammetry, Geometry of a single image, Characteristics of vertical imagery, Stereophotogrammetry, Aerial cameras and Photographs, Geometry of Aerial Photograph, Introduction to Ortho-photos and DEM/contour extraction, Aerial Photo Interpretation techniques and tools, Mapping from aerial photographs, Introduction to laser scanning, LiDAR data processing, classification, and segmentation.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, journal and conference papers, presentation files, and web resources
- The course will be delivered through class lectures, lab exercises, and tutorials

Prerequisite: CSE 3117

Reference Books and Journals:

1. Introduction to Remote Sensing, Fifth Edition – James B. Campbell.
2. ISPRS Journal of Photogrammetry and Remote Sensing
3. International Journal of Remote Sensing
4. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4280	**Introduction to Bioinformatics**	**Theory: 3**	**15 × (1×3) = 45**

Introduction of the Course:

This course is designed to give students both a theoretical background and a working knowledge of the state-of-the-art techniques employed in bioinformatics. Emphasis will be placed on biological sequence (both bulk and single-cell assays) analysis and its applications.

Specific Objective:

- Students will become familiar with a variety of currently available genomic and proteomic databases and will be able to search and retrieve inform from these databases.
- Students will learn how to compare and analyze biological sequences and how to interpret the results of their analyses.
- Students will learn the principles and applications of microarrays as well as single-cell technologies.
- Students will be able to perform elementary comparative genomic analysis, protein structure prediction, gene regulatory network reconstruction.

Course Contents: Essentials of Molecular biology: DNA, RNA and Protein, Watson and Crick Model of DNA, DNA replication, transcription, translation, splicing, Central dogma of molecular biology; DNA sequencing technology; Sequence Databases; Sequence Formats, Single-cell RNA-sequences, microarrays; Pairwise sequence alignment: local and global alignment, amino acid substitution scoring matrices; significance of the alignment, Multiple sequence alignment: progressive, iterative, statistical methods for multiple sequence alignment; local multiple alignment - sequence profiles and motifs, EM algorithm, MCMC algotithm, Gibbs Algorithm, Variational Inference; Hidden Markov Models: theory; training and applications to sequence alignment. Sequence database search: heuristic methods – FASTA, BLAST; Global Alignment methods – MatGat, Multiple Alignment using Clustal; Basic methods in molecular phylogeny: phylogenetic trees; distance matrix methods; maximum parsimony methods; maximum likelihood methods. RNA secondary structure basics: energy minimization, comparative sequence analysis, combined sequence and computational method; Gene Prediction: neural network, pattern discrimination functions and HMM; Promoter prediction: pattern driven, sequence driven algorithms; Proteins: protein classification, structure alignment and prediction; microarrays: design, data acquisition and analysis; gene network modeling; Systems biology.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite:

References:

1. Bioinformatics: Sequence and Genome Analysis by David W. Mount, Cold Spring Harbor Laboratory Press; 2001.
2. Bioinformatics: the machine learning approach by Baldi P, Brunak S., MIT press; 2001.
3. An introduction to bioinformatics algorithms by Jones, N.C. and Pevzner, P.A., MIT press; 2004.
4. Bioinformatics: genes, proteins and computers by Orengo, C., Jones, D. and Thornton, J. eds., Taylor & Francis; 2003.
5. Next-Generation Sequencing Data Analysis by Xinkun Wang, CRC Press; 2016.

Course No.	Course Title	No. of Credits	Credit Hours
CSE 4281	**Genomics Data Science**	**Theory: 3**	15 × (1×3) = 45

Introduction of the Course:

Genomic Data Science is the field that applies statistics and data science to the genome. The goal is to be able to better understand the genome and be able to leverage the data and information from genomic datasets. This course aims to cover the modern concepts and tools to understand, analyze, and interpret genome data from next generation sequencing as well as microarray experiments. It focuses on the most common tools used in genomic data science. This course is expected to serve as a standalone introduction to genomic data science as well as to help the students seeking to gain familiarity in data science and statistical tools to better interact with the data in their everyday work.

Specific Objective:

- Students will learn state-of-the-art techniques for analyzing genomics data.
- Students will learn how to structure, annotate, normalize, and interpret genome-scale assays.
- Students will learn how to analyze data from several experimental protocols using open source softwares.

Course Contents: Introduction to the statistics for genomics data science: Data normalization, exploratory analysis, linear modeling, testing, and multiple testing; preprocessing, batch effects, hypothesis testing, multiple hypothesis testing; Generative model for non-continuous outcomes – binary data, count data; mixture models, clustering, visualization; Multivariate analysis – high throughput genomics data, Multivariate methods for heterogeneous data, supervised learning, networks and trees. Scoring matrices: Protein and nucleotide scoring matrices i.e.PAM, BLOSUM, Gonett. How to construct scoring matrices. Difference between PAM and Blosum. Database homology search: Concepts behind BLAST: Applications & Biological Significance; homology, similarity & identity Statistical significance of BLAST: E value, Scores BLAST versions- BLASTp, BLASTn, Difference between FASTA and BLAST. Single-cell genomics analysis: Qualitiy control, preprocessing and normalization, Exploratory analysis – Cell type annotation and identification, trajectory inference and analysis. Phylogenetic analysis: Basic terminology in Phylogenetics: Distance and parsimony methods; Clustering methods. Rooted and un-rooted trees. Predictive methods using DNA sequences: Gene predictive methods-searching by signal, searching by content, homology based predictions, Markov models, Hidden Markov models in gene prediction: Genscan, Glimmer, Grail. Promoter analysis and predictions. Protein Structure Prediction: Secondary structure prediction methods: CHAU FASMAN, GOR, NN Tertiary Structure prediction methods- Homology Modeling, Threading/Fold recognition and Ab initio.

Instructional Strategies:

- The medium of instruction is English
- Lecture materials: recommended books, ppt files, and documents

Prerequisite:

References:

1. Modern Statistics for Modern Biology by Susan Holmes and Wolfgang Huber, Cambridge University Press; 2019.
2. Bioinformatics: Sequence and Genome Analysis by David W. Mount, Cold Spring Harbor Laboratory Press; 2001.
3. Bioinformatics: the machine learning approach by Baldi P, Brunak S., MIT press; 2001.
4. An introduction to bioinformatics algorithms by Jones, N.C. and Pevzner, P.A., MIT press; 2004.
5. Next-Generation Sequencing Data Analysis by Xinkun Wang, CRC Press; 2016.
6. Computational Genome Analysis: An Introduction (Statistics for Biology & Health S) by Richard C. Deonier, Simon Tavaré and Michael S. Waterman, Springer; 2005.

Full set to Bari
on 12.11.23 by [illegible]

Course Contents are Ready
to Bari [illegible]