Dept. of Computer Science and Engineering

Jahangirnagar University

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

(Effective from 2018-19)

Detail Syllabus

of

Forth Year First Semester

Course code	**:**	CSE 400	**Credit**	**:**	1.0
Title	:	Viva-Voce	**Prerequisite**	:	None
Type	:	*Viva-Voce*	**Contact hours**	:	-

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

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

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

Course Description

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

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

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

Rationale

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

Course Objectives

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

Students Learning Outcomes

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

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

Course Description

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

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

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

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

Rationale

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

Course Objectives

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

Students Learning Outcomes

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

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

Course Description

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

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

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

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

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

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

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

Lab Course Description

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

Hardware and Software Requirements

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

---

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

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

Course Objectives

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

Students Learning Outcomes

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

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

Course Description

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

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

Recommended Books

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

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

Rationale

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

Lab Objectives

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

Lab Outcome

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

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

Lab Course Description

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

Hardware and Software Requirements

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

---

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

Rationale

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

Course Objectives

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

Students Learning Outcomes

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

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

Course Description

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

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

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

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

Rationale

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

Lab Objectives

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

Lab Outcome

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

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

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

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

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

---

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

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

Course Objectives

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

Students Learning Outcomes

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

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

Course Description

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

References

The reading materials are provided by the supervisor.