**Dept. of Computer Science and Engineering**

Jahangirnagar University

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

(Effective from 2018-19)

**Detail Syllabus**

**Of**

**Third Year First Semester**

| | | | | | |
|---|---|---|---|---|---|
| Course code | : | CSE 300 | Credit | : | 1.0 |
| Title | : | Viva-Voce | **Prerequisite** | : | None |
| Type | : | *Viva-Voce* | **Contact hours** | : | - |

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

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

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

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

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

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

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

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

Students Learning Outcomes

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

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

| # | Topics and Descriptions |
|---|---|
| 1. | **Introduction** |

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

Recommended Books

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

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

---

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

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

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

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

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

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

Recommended Books

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

---

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

Rationale

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

Lab Objectives

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

Lab Outcome

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

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

- Study to create simple animations.

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

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

---

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

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

levels of resolution and how to cope with high dimensions.

Course Objectives

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

Students Learning Outcomes

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

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

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

Recommended Books

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

---

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

Rationale

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

Course Objectives

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

mechanisms; inter-computer communication concepts

Students Learning Outcomes

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

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

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

Recommended Books

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

---

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

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

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

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

| # | Topics and Descriptions |
|---|---|

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

Recommended Books

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

| | | | | | |
|---|---|---|---|---|---|
| | | J. Deitel, David R. Choffnes | | | |

---

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

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

Lab Objectives

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

Lab Outcome

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

- Compare the performance of various CPU Scheduling Algorithms.
- Implement deadlock avoidance, and Detection Algorithms.
- Critically analyze the performance of the various page replacement algorithms.

Lab Course Description

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

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

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

---

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

Rationale

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

Lab Objectives

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

- HyperText Markup Language (HTML) for authoring web pages

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

Lab Outcome

After completing this Laboratory course, the students will have:

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

Lab Course Description

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

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

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

---

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

Rationale

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

Lab Objectives

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

Lab Outcome

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

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

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

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

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