Syllabus Document

B. Tech. CSE (Human Computer Interaction & Gaming Technology)
2023 Batch Onwards

1st Semester

Course Code:	**MAL 103**	**Course Title:**	**Calculus for Engineers**			
Category:	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**1**	**0**	**4**
Prerequisite (if Any)	**NONE**	**Type of Course**	**Basic Sciences**			

Course Outcomes:

The students will be able

1. To analyze the nature (convergence or divergence) of a sequence or series.
2. To apply mean value theorems in the study of motion of an object.
3. To use integration in the calculation of area, volume, mass, and centre of gravity.
4. To apply multivariable calculus to study the nature of multivariable functions.
5. To exploit vector calculus in engineering problems.

Course Contents:

Module 1: Sequences and series: Sequences of real numbers, Series, ratio and root test.

Module 2: Calculus of functions of single variable: Review of limits, continuity, and differentiability. Mean value theorems: Rolle's theorem, Lagrange's theorem, Cauchy's theorem, Taylor's theorem with remainders, indeterminate forms, curvature, curve tracing.

Module 3: Fundamental theorem of Integral calculus, mean value theorems of integral calculus, evaluation of definite integrals, applications in area, length, volumes and surface of solids of revolutions, Improper integrals: Beta and Gamma functions, differentiation under integral sign.

Module 4: Calculus of Functions of Several Variables: Limit, continuity and differentiability of functions of several variables, partial derivatives and their geometrical interpretation, Tangent plane and normal line. Total differentiation, chain rules, Taylor's formula, maxima and minima, Lagrange's method of undetermined multipliers. Double and triple integrals, Jacobian, change of order of integration, change of variables, application to area, volumes, Mass, Centre of gravity.

Module 5: Vector Calculus: Scalar and vector fields, gradient of scalar point function, directional derivatives, divergence and curl of vector point function, solenoidal and irrotational motion. Vector integration: line, surface and volume integrals, Statement of Green's, Stoke's and Gauss divergence theorems (without proof) and their applications.

Text Books:

1. Huges-Hallett et al., Calculus: Single and Multi Variable, John-Wiley & Sons (USA), 3rd edition, 2003.
2. George B.Thomas, D.Weir and J.Hass, Thomas Calculus, Pearson, 12th edition 2010.
3. J. Stewart, Calculus, Thomson, 5th Edition, 2003 (Indian Edition).

References:

1. John Bird, Higher Engineering Mathematics, Elsevier Limited, 5th Edition, 2006.

Course Code	CSL106	Course Title	Introduction To Gaming			
Category	Core	Credit Assigned	L	T	P	C
			2	0	0	2
Prerequisite (If any)	NONE	Type of Course	Computer Science and Engineering			

Course Outcomes:

1. Understand basic principles of Game Design and Game Design Process.
2. Understand importance of standards for good quality code and testing and the basics of display technology, Software Development Kit (SDK), Application Programming Interface (API).
3. Understand basic design guidelines for gaming application, industry wide best practices and various ways in game to grabs inputs from various devices.

Course Contents:

Module 1: Core Design: What Is a Game? Games Aren't Everything. Games Mean Gameplay. Creating the Game Spec. Example Game Spec, Initial Design: The Beginning. Hardware Abstraction. The Problem Domain. Thinking in Tokens.

Module 2: Use of Technology: The State of the Art. Blue-Sky Research. Reinventing the Wheel. Use of Object Technology, Building Bricks: Reusability in Software, Initial Architecture Design: The Birth of Architecture. The Tier System. Architecture Design.

Module 3: Development: The Development Process. Code Quality. Coding Priorities. Debugging and Module Completion. The Seven Golden Gambits. The Three Lead Balloons. GAME PROGRAMMING: Technologies: Display, Mixing 2D and 3D, DirectX, User Interface code, Resource caching, the main loop.

Module 4: Design Practices: Smart & naked pointers, using memory correctly, Game scripting languages, Building your game: Creating a project, source code repositories and version control, Building the game and scripts, User interface programming and input devices: Getting the Device State, Working with the Mouse (and Joystick), Working with the Keyboard, User Interface Components, More Control Properties.

Text Books:

1. Game Architecture and Programming, Shankarmani, Jain, Sinha, Wiley Publication, India
2. Fundamentals of Game Design, 3rd Edition, Ernest Adams, Pearson Publication

Reference Books:

1. Game Theory: An Introduction, E. N. Barron, Wiley Student Edition.
2. ActionScript 3.0 Game Programming University, 2nd Edition, Gary Rosenzweig, Pearson Education.
3. "Game Architecture and Design", Andrew Rollings and Dave Morris 4. "Professional Game Programming" Mike McShaffry, Dreamtech Press.

<table>
<tr><td>Course Code:</td><td>CSL 101</td><td>Course Title:</td><td colspan="4">Computer Programming</td></tr>
<tr><td>Category:</td><td>Core</td><td>Credit Assigned</td><td colspan="4"><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>2</td><td>4</td></tr></table></td></tr>
<tr><td>Prerequisite (if Any)</td><td>NONE</td><td>Type of Course</td><td colspan="4">Computer Science and Engineering</td></tr>
<tr><td colspan="6">Course Outcomes:
1. Recognise basics of programming and develop logical thinking.
2. To illustrate how to model real world problems into the software and develop practical programming skills.
3. To use mathematical and statistical applications into programming.
4. To analyse and develop solutions for the general as well as scientific problems.</td></tr>
<tr><td colspan="6">Course Contents:

Module 1:
Introduction – Computer generation and evolution, flowcharts, algorithm, What is C?, constants, variables, scope of variable, data types, operators, arithmetic expression, Hierarchy of operators, control flows, conditional operator, loops, switch concept.
Program Structure – Basic programs to illustrate structure of C program and its flow in execution.

Module 2:
Function – Introduction to function and parameter passing, returning value, recursive functions, macros.

Module 3:
Arrays – One-dimension and multi-dimension arrays, array initialization, how arrays are stored in memory, array as parameter in functions, programs based on arrays.

Module 4:
Pointers – Initialization, accessing a variable through pointers, pointers as function arguments, pointer to array, arrays of pointers, pointers to pointers.

Module 5:
Structure and Union – Defining a structure, accessing structure members, Array of structure, unions.

Module 6:
File Handling- reading from and writing to a file.</td></tr>
<tr><td colspan="6">Text books:
1) The C Programming Language by Brian W. Kernighan and Dennis M. Ritchie, PHI.
2) Programming in C by E. Balguruswamy, Tata Mcgraw Hill Publishing.</td></tr>
</table>

List of Lab Assignments / Experiments

1. Programs using function.
2. Programs using arrays.
3. Programs on structures.
4. File Handling

Course Code:	**BEL 101**	**Course Title:**	**Mechanics and Graphics**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **3** **0** **2** **4**
Prerequisite (if Any)	**NONE**	**Type of Course**	**Engineering Sciences**

Course Outcomes:

1. Use of various drawing instruments, Concept of scales, Representative factor and dimensioning, Orthographic projections of points, lines, plane on principle planes/ Profile plane/ Auxiliary planes. Projection of right regular solids inclined to both the planes. Projection of right regular solids inclined to both the planes. Drawing isometric views from orthographic projection orthographic views.
2. Principles of Vector representation of force system, Moment of a force about a point and about an axis; couple moment; reduction of a force system to a force – a couple Wrench , Free Body Diagram, Reactions at supports, Equilibrium of Planar (including friction) and Spatial force system,
3. Internal forces in member: Determination of variation of Axial force (Axial Force Diagram), Shear force (Shear Force Diagram), Bending moment (Bending Moment Diagram) and twisting moment (Torque diagram)
4. Concept of stress and strain: Normal and shear stress and strain, State of stress at a point, Stress strain curve, Hook's law, Modulus of elasticity, Poisson's ratio, Modulus of rigidity, Bulk modulus, Transformation of stress.

Course Contents:

Module I

Engineering Graphics

Engineering curves like cycloid, conic sections. Concept of scales, Representative factor. Orthographic projections of points, lines, plane. Projection of right regular solids inclined to both the planes. Conversion of isometric view to orthographic views. Isometric views. Development of surfaces. Introduction to CAD, applications, Softwares, AUTOCAD, Basic commands and problems in 2D and 3D.

Module II

Applied Mechanics

Principles of Vector representation of force system, Moment of a force about a point and about an axis; couple moment; reduction of a force system to a force – a couple Wrench , Free Body Diagram, Reactions at supports, Resultant and Equilibrium Analysis, Equilibrium of Planar and Spatial force system, friction. Internal forces in member (TRUSS): Determination of variation of Axial force (Axial Force Diagram), Shear force (Shear Force Diagram), Bending moment (Bending Moment Diagram) and twisting moment (Torque diagram). Concept of stress and strain: Normal and shear stress and

strain, State of stress at a point, Stress strain curve, Hook‟s law, Modulus of elasticity, Poisson‟s ratio, Modulus of rigidity, Bulk modulus, Transformation of stress.
Kinetics of Particles (a) D'Alembert's principle applied to bodies having rectilinear motion. (b) Principle of work and Energy: General numerical applications (c) Principle of Impulse and momentum: General numerical applications
Centre of Gravity, Second moment of area, polar moment of inertia, radius of gyration, Parallel axis theorem, Product of inertia.

Text:

1. Singer F.L. and Andrew Pytel, Strength of Material, Harper and Row Publishers, New York.
2. Bhatt N.D. and Panchal V.M., Elementary Engineering Drawing, Charotar Publishing House, 43rd edition.

Reference:

1. Hibbler, Engineering Mechanics, Pearson Education, Asia Pvt Ltd.
2. Beer F.P. and Johnston E.R., Vector Mechanics for Engineers: Statics and Dynamics, Tata McGraw-Hill
3. Irving H. Shames, Engineering Mechanics: Static and Dynamics, Pearson Education, Asia Pvt Ltd.
4. Meriam J.L. and Kraige L.G., Engineering Mechanics, John Wiley and Sons.
5. Stephen Timoshenko, Strength of Materaials, Part -1, CBS Publishers and Distributors, New Delhi.
6. Popov E.P., Mechanics of deformable bodies, Prentice-Hall
7. Beer F.P. and Johnston E.R., Mechanics of materials, McGraw-Hill International

<table>
<tr><td>Course Code:</td><td>SAP 101</td><td>Course Title:</td><td colspan="4">Health, Sports & Safety</td></tr>
<tr><td>Category:</td><td>Core</td><td>Credit Assigned</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>Prerequisite (if Any)</td><td>NONE</td><td>Type of Course</td><td colspan="4">Humanities</td></tr>
</table>

Course Outcomes:

1. To provide physical fitness and good health.
2. Create awareness among the students about their health status by conducting various tests and measurements and suggest them suitable remedial physical fitness program so that they can improve physical and physiological health status.
3. To improve productivity, foster social harmony, inculcate sense of discipline and dedication in general life, develop the spirit of team work, through various sports activities.

Course Contents:

Development of components of fitness through conditioning exercises: Strength: (Strength Endurance, Maximum Strength, explosive strength), Endurance: (aerobic endurance, anaerobic endurance, speed endurance and strength endurance), Speed, Co-coordinative ability, Flexibility

Physical Efficiency Test Level 1(Testing and Evaluation of Physical Fitness): Cooper Test 12 minute run or walk test, Sit and reach test, 100 meter run, one minute sit up test, Push up/Bent knee push up test

Teaching and development of sports skills: Cognitive, Perceptual, Motor, Perceptual motor. First Aid training

Intramural phase 1: Identification of sports talent through exposing students to inter- section tournament. Football, Volleyball, throw ball, table tennis & Chess.

Yoga, Meditation and Personal Safety.

Course Code:	**HUL 101**	**Course Title:**	**Communication Skills**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **2** **0** **2** **3**
Prerequisite (if Any)	**NONE**	**Type of Course**	**Humanities**

Course Outcomes:

After the successful completion of the course, the students will be able to:

1. Utilize functional English grammar for accurate and enhanced language skills.
2. Construct and use effective interpersonal and workplace communication
3. Acquire better reading comprehension, pronunciation and reading skills
4. Introspect and illustrate the personality traits and soft skills
5. Develop the skills for better pre and post placement communication through effective presentations, personal interviews and group discussions

Course Contents:

Module 1
Communication:-
Definition of Communication, Process of Communication, Stages of Communication, Content of the message, Types of communication, Transmission, Medium/Modes of Communication, Verbal and Non-verbal Communication (Kinesics, Proxemics, Chronemics, Haptics, Paralinguistic Feature), Levels of Communication, Flow of Communication, Communication Networks, Grapevine, Barriers to Communication, Choice of Medium,

Module 2
Listening Skills:-
Art of Listening, Listening vs Hearing, (Poor Listening vs Effective Listening), Advantages of Good Listening, Barriers to Effective Listening, Techniques of Effective Listening
Reading Skills:-
Reading Comprehensions, Process of Reading, Techniques of Reading, Techniques for Good Comprehension, Reading Skills(Skimming, Scanning, Intensive Reading, SQ3R), Orientation in Literary and Scholarly Article

Module 3
Speaking Skills:-
Types of Speech, Public Speaking, Components of Effective speech, Stage Presence & Personality Development, Clarity and Fluency, Body Language, , Barriers to Effective Speaking
Presentation Skills:-

Characteristics of a Successful Presentation, Power Point Presentation, Using Audio Visual Aids

Module 4
Group Discussion:-
Do's and Don'ts of GD, Essential Skills for GD, Evaluation Pattern
Personal Interview:-
Objectives of Interview, Types of Interview, Job Interviews, Employer's Expectations, Do's & Don'ts of Social Media Profile, Success Factors, Failure Factors

Module 5
Grammar:-
Transformation of Sentences, Punctuation, Spellings and Mechanics of Writing

Text Books:

1. Orient Longman, A Textbook of English for Engineers and Technologists
2. M. Ashraf Rizvi, Effective Technical Communication. Tata Mc Grwa-Hill Publishing Company Limited, 2009

Reference Books:

1. Quirk R. and Greenbaum S., A University Grammar of English.
2. Krishnaswamy N., English Grammar (Longman Publication) (Macmillan India Ltd)
3. Sanjay Kumar and PushpaLata, Communication Skills. Oxford Publication
4. Meenakshi Raman and Sangita Sharma. Technical Communication. Second Edition. Oxford Publication,2011

List of Lab Assignments/Experiments

1. Speaking Skills (Verbal/Non verbal Skit, Role Play, Extempore, Story Telling, Word Wheel, Debate)
2. Presentation Skills (Film/Book Review, PPT Presentation)
3. Group Discussion (Practice GD, Mock GD)
4. Personal Interview/ SWOT Analysis (SWOT Analysis, Mock PI)
5. Comprehending a Technical Report/News Paper Article.
6. Presenting a Book Chapter using PowerPoint slides

Course Code	**CSL107**	**Course Title**	**Introduction to HCI**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			2	**0**	**0**	**2**
Prerequisite (If any)	**NONE**	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

1. Understand the Principles of HCI
2. Apply User-Centered Design Methods
3. Evaluate and Critique Interfaces
4. Design and Develop Usable Interfaces

5. Understand Ethical and Social Implications

Course Contents:

Module 1: What is HCI?, Disciplines involved in HCI, Why HCI study is important? The psychology of everyday things, Principles of HCI, User-centred Design.

Module 2: Input-output channels, Human memory, Thinking: Reasoning and Problem Solving, Human emotions, Individual differences, Psychology and Design.

Module 3: Models of interaction, Ergonomics, Interaction styles, WIMP Interface, Interactivity, Context of interaction, User experience, Paradigms of Interactions.

Module 4: What is interaction design?, The software design process, User focus, Scenarios, Navigation Design, Screen Design, Prototyping techniques, Wire-Framing, Understanding the UI Layer and Its Execution Framework

Module 5: Introduction to : Principles that support usability, Design standards, Design Guidelines, Golden rules and heuristics, Using toolkits, User interface management system (UIMS), Goals of evaluation, Evaluation Criteria, Evaluation through expert analysis, Evaluation through user participation, Choosing an Evaluation Method.

Module 6: Introduction to: Goal and task hierarchy model, Linguistic model, Physical and device models, Cognitive architectures, Hierarchical task analysis (HTA), Uses of task analysis, Diagrammatic dialog design notations, Computer mediated communication, Ubiquitous Computing, Finding things on web Future of HCI.

Text Books:

4. Alan Dix (2008). Human Computer Interaction. Pearson Education. ISBN 978-81-317-1703-5.

2nd Semester

Course Code:	**MAL 104**	**Course Title:**	**Matrices, Transform Techniques, and Differential Equations**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **3** **1** **0** **4**
Prerequisite (if Any)	**MAL 103**	**Type of Course**	**Basic Science**

Course Outcomes:
The students will be able

1. To apply the concepts of matrices for solving system of linear equations.
2. To use the Fourier transform and Laplace transform for solving the differential equations.
3. To solve different types of differential equations.
4. To model the LR-circuit, RL-circuit, radioactive decay, and population growth, and heat equation.
5. To exploit partial differential equations in solving heat flow problems.

Course Contents:

Module 1: Matrices:Vectors in R^n, notion of linear independence and dependence, rank of matrix, system of linear equations, Gauss-elimination, eigenvalues and eigenvectors, algebraic and geometric multiplicity, Cayley – Hamilton theorem, linear and orthogonal transformations, reduction to diagonal form, Hermitian and skew Hermitian matrices. Applications of linear transformations in computer graphics.

Module 2: Introduction to complex numbers and domain transformation: Review of complex numbers, Fourier series, Dirichlet conditions, half range series, RMS value, Parseval's identity, Fourier transform and properties, Fourier sine and cosine transforms, Convolution Theorem and Parseval's identity, Simple problems, Definition of Laplace transform-Properties-Laplace transform of periodic functions, Laplace transform of unit step function, impulse function, inverse Laplace transform, convolution, Z-transform, Z-transforms of standard functions, inverse Z-transforms: by partial fraction method and convolution method.

Module 3: Ordinary Differential Equations: First order differential equations: exact equations, integrating factors and Bernoulli's equations, orthogonal trajectories, existence and uniqueness of solutions. Solutions of second and higher order differential equation with constant coefficients: homogeneous and non-homogeneous. Linear independence and dependence of solutions, method of variation of parameters, Cauchy-Euler equations, simultaneous linear equations.

Module 4: Applications in LR-circuit, RL-circuit, radioactive decay, and population growth, LCR circuit and vibrations in electrical systems.

Module 5: Partial Differential Equations: Formation and solution of partial differential equations, General, particular, complete and singular integrals, partial differential equations of first order of the forms: F(p,q)=0, F(z,p,q)=0, F(x,p)=G(y,q) and Clairaut's form, Lagrange`s equation: Pp+Qq = R.Linear partial differential equations of higher order with constant coefficients. Classification of second order PDE, solution of a partial differential equation by separation of variables, boundary value problems-one dimensional wave and heat equations-heat flow in an infinite bar, wave propagation on a semi infinite string – one dimensional heat equations in steady state – solution using Fourier transforms.

Text Books:

1. Erwin Kreyszig , Advanced Engineering Mathematics, John WileyIndia, 10th Edition, 2015
2. Gilbert Strang, Introduction to Linear Algebra, Wellesley- Cambridge Press, 4th Edition, 2011.
3. G. F. Simmons, Differential Equations with Applications and Histrorical Notes, Tata McGraw Hill, Second Edition, 2003.
4. J. D. Logan, Applied Partial Equations, Springer-Verlag, 3rd Edition, 2015.
5. S. Kumaresan, Geometrical Approach to Linear Algebra, Prentice Hall of India, 2000.

Reference:

1. Jin Ho Kwak and Sungpyo Hong, Linear Algebra, Springer, Second edition, 2004.
2. S. L. Ross, Partial Differential Equations, Wiley, 3rd Edition, 2007.

Course Code	**ASL103**	**Course Title**	**Applied Physics for Gaming**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Prerequisite (If any)	**-**	**Type of Course**	**Basic Science**			
Course Outcomes:						

1. To identify the components of real-time game physics simulation and to analyse different type of Newtonian Mechanics for game.
2. To develop the fundamental knowledge of projectile and to use kinematic equations to analyse and solve horizontally and angled launched projectile.
3. To analyse game objects and understand the corresponding collision impulse and momentum principle to produce the desired interactive effect in game.
4. To analyse and compare gaming mechanics of cars, motorcycles, boat, buoyancy and things that float.
5. To analyse the external physics of Airplanes, Rockets and Missiles

Course Contents:

Module I: Newtonian Mechanics and Basic Kinematics

Basic concepts Game physics - Physics realism –importance in games, physics concepts and game performance.

Basic Newtonian mechanics & Kinematics Newtons three laws of motions – inertia – force – mass – acceleration, equal& opposite forces, Force vector, Types of forces – gravitational – friction-centripetal – force balance and diagrams, Work, Energy: kinetic – potential – conservation – power, Translational motion – equation of motion, Rotational motion - torque – angular acceleration, 2D particle kinematics, 3D particle kinematics, Rigid body kinematics – centre of mass – coordinate axis – rolling motion –- bowling ball kinematics, Exercise and problem solving.

Module II: Projectiles

Projectile's properties, the gravity only Model-Simple trajectories & gravity, Aerodynamic Drag -d– drag coefficient –drag effects and equation of motion, Wind effects to projectile, Spin effects: Magnus effect – spin effect to projectile, specific projectiles types in games – cannon balls – bullets – arrows, Exercise and problem solving.

Module III: Collisions

Collisions Impulse and momentum principle- linear and angular, elastic & inelastic collisions Impact, coefficient of restitution, collision direction and detection, collision with movable & immovable objects, collision with friction, 2D and 3D collisions, Exercise and problem solving.

Module IV: Sports Simulations: Cars, Motorcycles and Boats and Things That Float

External physics of gaming vehicles (motorcycles, cars, ships) - Cars motorcycle: Engine Torque and Power, Gear Shifting, Manual and Automatic Transmissions, Driving Around Curves. Boats and Things that float: Basic Force Diagram of boats and hulls, Buoyancy, Density, Drive System Types, Propeller Basics, Skin Friction Drag, Form Drag, Powerboat Turns, Physics of Sailing, Physics of Surfing, Buoyancy and Balance, Catching a Wave.

Module V: Airplanes, Rockets and Missiles

External physics of gaming vehicles (airplanes, rocket and missiles), Airplanes: Airplane Terminology, Lift and its evaluation using Air foils, flaps, center of pressure, Thrust, Jet Engines, Skin Friction and Form Drag, Induced Drag, Total Drag Equation and full-body aerodynamics, Aircraft orientation, Trim and Stability, Dynamic Stability.

Rockets and Missiles: Different Types of Engines, Thrust, Specific Impulse, Altitude Effects, Drag effects for rockets at different altitudes, Circular and Elliptical Orbits, Escape Velocity, Missiles and Multi-stage Rockets.

Text Books:

1. Physics for Game Programmers, Grant Palmer, Apress publishers, (2005)
2. Fundamental Physics, Halliday – Resnick, 8th Edtion, Weiley (2009)
3. Game Physics, David H. Eberly, Morgan Kaufmann publications, Second Edition (2010)
4. Classical Mechanics, Herbert Goldstein, 3rd Edition, Addison- Wesley (2002)

Reference Books:

1. Game Physics Engine Development, Ian Millington, Morgan Kaufmann Publishers, 2 n d Edition, CRC press, 2010
2. Mathematics and Physics for Programmers, John Patrick Flynt& Danny Kodicek, Second Edition, Course Technology, a part of Cengage Learning. (2012).
3. Mechanics, Keith R. Symon, 3rd Edition, Addison- Wesley, (1971)
4. Advanced Physics, M. Nelkon, P. Parker, 7 th Edition, Heinemann Educational Books (1970).

Course Code:	**CSL 102**	**Course Title:**	**Data Structures**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **3** **0** **2** **4**
Prerequisite (if Any)	**CSL 101 (Computer Programing)**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

Students will be able to:

1. Design and differentiate the recursive and iterative versions of the program.
2. Implement and analyse algorithms using dynamic memory allocation.
3. Apply, implement and evaluate various concepts of linear data structures for solving real life problems.
4. Apply, implement and evaluate non-linear data structures in solving various problems.

Course Contents:

Module 1:

Types and operations, Iterative constructs and loop invariants, Quantifiers and loops, Structured programming and modular design, Illustrative examples, Scope rules, parameter passing mechanisms, recursion, program stack and function invocations including recursion, Overview of arrays and array based algorithms - searching and sorting, Mergesort, Quicksort, Binary search, Introduction to Program complexity (Big Oh notation)

Module 2:

Implementation of Structures (Records) and array of structures (records). Database implementation using array of records. Dynamic memory allocation and deallocation. Dynamically allocated single and multi-dimensional arrays.

Module 3:

Lists as dynamic structures, operations on lists, implementation of linked list using arrays pointers and its operations. Introduction to Overview of linked list implementation using self-referential-structures/pointers. Stack, Queues and its operations. Concept of an Abstract Data Type (ADT),Implementation of stacks and queues using both array-based and pointer-based structures. Uses of stacks in simulating recursive procedures/ functions. Applications of stacks and queues.

Module 4:

Lists - Singly-linked lists, doubly linked lists and circular linked lists. List traversal, insertion, deletion at different positions in the linked lists, concatenation, list-reversal etc. Mergesort for linked lists.

Module 5:

Trees, binary trees, binary trees- basic algorithms and various traversals. Binary Search Trees (BSTs) and insertion, deletion in BSTs.

Module 6:
Generalization of trees to graphs – their representation & traversals. Directed Acyclic Graphs and topological sort, Dijkstra‟s shortest path algorithm,

Text:
1. Data Structures & Program Design in C: Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.
2. Fundamentals of Data Structures in C : E. Horowitz, S. Sahni, and S. Anderson-Freed, University Press

Reference:
1. Aho, Hopcroft and Ullmann, ―Data Structures and Algorithms,‖ Addison Wesley, 1983.

Course Code:	**CSL 103**	**Course Title:**	**Application Programming**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **2** | **4**
Prerequisite (if Any)	**CSL 101 (Computer Programming)**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:
1. To recognize and use different tools for Web Programming.
2. Analyse the working background of web.
3. Construct efficient web pages with CSS and Javascript.
4. Demonstrate competency in the use of common HTML code for the development of website.

Course Contents:
Internet fundamentals, LAN, WAN, Introduction to common Internet terms, www. Basics of networking, DNS, URL, firewall, proxy, Web protocols – http and https.

Designing web pages: HTML, forms, DHTML, XML, CSS. Extensible Hypertext Mark up Language (XHTML): XHTML syntax, headings, linking, images, special characters and horizontal rules, lists, tables, forms, internal linking, meta elements.

Introduction to Web Server – Setting up and configuration of Apache Tomcat server, Accessing pages from another machine.

Server Side Programming: Introduction to web programming with PHP. Client side programming with Javascript.

Introduction to Python - Statements and Control Flow, Expressions, Methods, Typing, Libraries and Developmental Environment, Web Programming using Python.

Text:
1. Deitel H.M. and P. J. Deitel, Internet & World Wide Web - How to Program, Prentice-Hall.
2. Goodman D, Morrison M., JavaScript Bible; Wiley India
3. Lutz, Mark, Learning Python (4th ed.). O'Reilly Media

Reference:
1. Garfinkle S., Spafford G; Web Security, Privacy and Commerce; O'Reilly, 2002.
2. Atkinson L., Core PHP Programming, Prentice Hall.

3. N.P.Gopalan, Akilandeswari, Web Technology, Prentice-Hall.

Course Code	CSL108	Course Title	Game Development Design Thinking			
Category	Core	Credit Assigned	L	T	P	C
			2	0	0	2
Prerequisite (If any)	-	Type of Course	Computer Science and Engineering			

Course Outcomes:

1. To analyse the various design techniques
2. To develop design methodology by using different technique
3. Apply reverse engineering to determine the construct of product
4. To draw the technical drawing for the design of game

Course Contents:

Module 1:

Design Process: Product Life Cycle, Design Ethics, Design Process - Four Step, Five Step, Twelve Step, Creativity and Innovation in Design Process, Design limitation.

Module 2:

Creating and Developing Ideas: Creative Thinking, Generating Design Ideas, Lateral Thinking, Analogies, Brainstorming, Mind mapping, Naminal Group Technique, Synectics, Development of work, Analytical Thinking, Group Activities Recommended.

Module 3:

Reverse Engineering: Reverse Engineering for New Understanding about Products, Reasons for Reverse Engineering, Reverse Engineering Process, Step by Step - Case Study.

Module 4:

Drawing to develop design ideas: Many Uses of Drawing, Communication through Drawing, Drawing Basis: Line, Shape/ Form, Value, Colour, Texture.

Module 5:

Technical drawing to develop design: Perspective Drawing - One Point Perspective, Two Point Perspective. Isometric Drawing, Orthographic Drawing, Sectional Views.

Text Books:

1. Everett N McKay, UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication, 2013.
2. Don Norman, The Design of Everyday Things: Revised and Expanded Edition, 2013.
3. Steve Krug, Don't Make Me Think: A Common Sense Approach to Web Usability, 2nd ed., 2005
4. Jesse James Garrett, The Elements of User Experience: User-Centered Design for the Web and Beyond, 2nd ed, 2010.

Reference Books:

1. Russ Unger, A Project Guide to UX Design: For user experience designers in the field or in the making, 2nd ed., 2012.
2. Jeff Johnson, Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines, 2014
3. Chris Nodder, Evil by Design: Interaction Design to Lead Us into Temptation, 2013.
4. Jon Yablonski, Laws of UX: Using Psychology to Design Better Products & Services 1st Edition, 2020.
5. Andrew Couldwell, Laying The Foundations: How to Design Websites and Products Systematically, 2020.

Course Code:	**HUL 102**	**Course Title:**	**Environmental Studies**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **2** **0** **0** **2**
Pre-Requisite (if Any)	**NONE**	**Type of Course**	**Humanities**

Course Outcomes:

1. Identify natural resources, ecosystem, and biodiversity, their structure and functions.
2. Describe the importance of environmental components, and their role in human life.
3. Illustrate the possible causes of various forms of environmental pollution, their consequences, and methods of prevention.
4. Define the concept of sustainable development and mechanism to attain it.
5. Recognize the integration of social issues and environmental problems.

Course Contents:

Module 1: Natural resources: Forest resources, Water resources, Mineral resources, Food resources, Energy resources, Land resources.

Module 2: Ecosystem: Concept of an ecosystem, Structure and functions of an ecosystem, Producers, consumers and decomposers, Ecological succession, Food chain, food webs and pyramids.

Module 3: Biodiversity and its conservation: Introduction, definitions: genetics, species and diversity, Value of biodiversity, Biodiversity at global, national and local level, India as a mega-diversity nation, Hot-spot of biodiversity, Threat to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, Conservation of biodiversity: in-situ and ex-situ conservation.

Module 4: Environmental pollution: Definition, Causes, effects and control measures of: Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards. Solid waste management: Causes, effects and control measures of urban and industrial wastes.

Module 5: Social issues and environment: Sustainable development, Water conservation, Rain water harvesting, Watershed management, Climate change, Global warming, Ozone layer depletion, Nuclear accident, Environmental rules and regulations, Human population and environment. Role of information technology in environment and human health.

Case studies related to ecosystem, environmental pollution, sustainable development will be discussed in a class.

Text:

1. Rajagopalan, Raghavachari. *Environmental studies: from crisis to cure*. No. Ed. 3. Oxford University Press, 2015.

Reference:

1. Joseph, Benny. *Environmental studies*. Tata McGraw-Hill Education, 2005.
2. Chopra, Kanchan and G. K. Kadaekodi, Operationalizing Sustainable Development: Economic- Ecological Modeling for Developing Countries: Sage, Chapter 1, 1999.
3. Kolstad, Charles, D., *Environmental Economics*, Press, 2000.
4. Reed, David (Ed.), *Structural Adjustments, the Environment and Sustainable Development*, Earthscan, Chapters 1, 12, 13 and 14, 1996.
5. Bharucha, Erach. *Textbook of Environmental Studies for Undergraduate Courses*. Universities Press, 2005.

Course Code	**ECL103**	**Course Title**	**Applied Electronics**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (If any)	-	**Type of Course**	**Engineering Sciences**			

Course Contents:

Module 1: ELECTRONIC DEVICES

Theory of P-N Junction Diode, Junction Transistors Theory of Operation, Static Characteristics, Break Down Voltages, Current Voltage Power Limitations, Field Effect Transistor & MOSFET, Principle of Operation & Characteristics.

Module 2: APPLICATIONS of ELECTRONIC DEVICES

Rectifiers, Zener Diode as Regulators, Biasing of BJT Different Biasing Arrangements, Stability Factor, Small Signal Analysis & High Frequency Analysis of BJT, Power Amplifiers, Push Pull Configuration, Complimentary Symmetry, Feedback Amplifiers, RC, LC & Crystal Oscillators.

Module 3: COMBINATIONAL and SEQUENTIAL LOGIC

Logic minimization using K-map method, multiplexers, demultiplexers, decoders, encoders, Arithmetic circuits, Adders, Combinational multiplier and code converters. Basic latches, master-slave latch, Flip flops, Registers, Counters.

Module 4: MEMORIES

Introduction to PLA, PAL and ROM, Programmable Logic Devices and FPGAs.

Module 5: INTRODUCTION TO MICROPROCESSORS

Architecture, bus structure, timing diagrams, T-states, machine cycle, instruction cycle. Memory and IO devices interfacing.

Reference Books:

1) Electronic devices and circuit theory / Robert L. Boylestad, Louis Nashelsky
2) Milman and Halkias, “Integrated Electronics”, Second Edition, 2011, McGraw Hill.
3) Digital Design by M. Morris Mano and Michael D. Ciletti

4) Microprocessor Architecture, Programming, and Applications with the 8085 by Ramesh Gaonkar

2nd Year Syllabus

3rd Semester

<table>
<tr><td>Course Code:</td><td>CSL 204</td><td>Course Title:</td><td>Discrete Maths and Graph Theory</td></tr>
<tr><td>Category:</td><td>Core</td><td>Credit Assigned</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>1</td><td>0</td><td>4</td></tr></table></td></tr>
<tr><td>Prerequisite (if Any)</td><td>None</td><td>Type of Course</td><td>Computer Science and Engineering</td></tr>
<tr><td colspan="4">Course Outcomes:
1. Student should be able to use different proof techniques.
2. Students would be able to argue about limits by using Pigeon Hole principle.
3. Solve problems based on set theory, Permutations and Combinations, as well as Discrete Probability.
4. Students will be able to solve mathematical problems on partial orders, and group theory.
5. Students would be able to model and analyze computational problems in graph theoretical framework.</td></tr>
<tr><td colspan="4">Course Contents:

Module 1:
Set Theory: Definition of sets, countable and uncountable sets, Venn Diagrams, proofs of some general identities on sets Relation: Definition, types of relation, composition of relations, Pictorial representation of relation, Equivalence relation, Partial ordering relation, Job-Scheduling problem.

Module 2:
Algebraic Structures: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results, Rings and Fields: definition and standard results.

Module 3:
Propositional Logic: Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Normal Forms, Universal and existential quantifiers. Introduction to finite state machine Finite state machines as models of physical system equivalence machines, Finite state machines as language recognizers.

Module 4:
Graph Theory: Introduction and basic terminology of graphs, Planer graphs, Multigraphs and weighted graphs, Isomorphic graphs, Paths, Cycles and connectivity, Shortest path in weighted graph, Introduction to Eulerian paths and circuits, Hamiltonian paths and circuits, Graph coloring, chromatic number, Isomorphism and Homomorphism of graphs. Graphs, hypergraphs, transitive closure, trees, spanning trees. Eulerian tours, Hamiltonian cycles, Planar Graphs, Connectivity, Colorability, Line Graphs</td></tr>
</table>

Module 5:
Posets, Hasse Diagram and Lattices: Introduction, ordered set, Hasse diagram of partially, ordered set, isomorphic ordered set, well ordered set, properties of Lattices, bounded and complemented lattices.
Combinatorics: Introduction, Permutation and combination, Binomial Theorem, Multimonial Coefficients Recurrence Relation and Generating Function, Recurrence Relation and Recursive algorithms , Linear recurrence relations with constant coefficients, Homogeneous solutions, Particular solutions, Total solutions

Text Books:
1. C.L.Liu, "Elements of Discrete Mathematics" Tata Mc Graw-Hill Edition.
2. Trembley, J.P & Manohar; "Discrete Mathematical Structure with Application CS", McGraw Hill.
3. Kenneth H. Rosen, "Discrete Mathematics and its applications", McGraw Hill.
4. Lipschutz; Discrete mathematics (Schaum); TMH
5. Deo, Narsingh, "Graph Theory With application to Engineering and Computer Science.", PHI.

Course Code	**CSL211**	**Course Title**	**Gamification for Learning**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**0**	3
Prerequisite (If any)	-	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

Student will able to do:

1. To conceptualization the gamification and its usages in different contexts.
2. Identify major patterns of relevant case studies and summarize them effectively.
3. Apply management approach towards problem-solving.

Course Contents:

Module 1:

Why Games, Gamification, and Simulations for Learning? Game, Gamification, or Simulation: Which Is Best, When, Why? Critical Questions for Creating an Interactive Learning Event, Engagement and Fun.

Module 2:

Foundational Elements, The Importance of Narrative/Context/Story, Engagement and Flow Theory, User Perception of Gamification.

Module 3:
Where to Find Ideas, Games, Gamification, Simulations, Storyboarding.

Module 4:

Gamification Theory, Gamification Framework, Gamification Strategy, Legal and Ethical Issues, Effects of Gamification.

Module 5:
The Knowledge Guru, A Board Game: MPE, Mobile Gamification: Mobile Cricket U, Serious Game: Learning to Negotiate, Structural Gamification for On-Boarding Employees, Medical Simulation, Financial Game-Based Learning, Sales Training Game: An Avaya Case.

Text Books:
1. Karl M. Kapp, "The Gamification of Learning and Instruction Field book: Ideas into Practice", Wiley, 2013.
2. Sangkyun Kim, "Gamification in Learning and Education: Enjoy Learning Like Gaming", Springer, 2017.

Reference Books:

1. Yu-Kai Chou, Actionable Gamification. Beyond Points, Badges, and Leaderboards, Freemont (CA), 2014.

Course Code:	**CSL202**	**Course Title:**	**Introduction to Object Oriented Programming**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **3** **0** **2** **4**
Prerequisite (if Any)	**None**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:
Students will be able to:
1. Evaluate various object-oriented paradigms like abstraction, encapsulation, inheritance, polymorphism, information hiding, exception handling, etc.
2. Identify the implementational differences between different object-oriented programming languages.
3. Analyse a problem description and design and develop object-oriented software using best coding practices.
4. Assess object-oriented solutions for various real-world problems.

Course Contents:
Module 1:
Object Oriented Programming, Features of object oriented programming languages like data encapsulation, inheritance, polymorphism and late binding.

Module 2:
Concept of a class, Access control of members of a class, instantiating a class, static and non-static members, overloading a method. Deriving a class from another class, access control of members under derivation, different ways of class derivation, overriding of a method, run time polymorphism.

Module 3:
Concept of an abstract class. Concept of an interface. Implementation of an interface. Exception and exception handling mechanisms. Study of exception handling mechanisms in object-oriented languages

Module 4:

Introduction to streams, use of stream classes. Serialization and de-serialization of objects. Templates, Implementation of data structures like linked lists, stacks, queues, trees, graphs, and hash table etc. using object oriented programming languages.

Module 5:
Introduction to concept of refactoring, modelling techniques like UML, Design patterns.

Text:

1. Bjane Strostrup, "The C++ programming language", Addison-Wesley
2. Herbert Shildt, "C++: The Complete Reference", 4th Edition
3. Arnold Ken, Gosling J, "The Java Programming Language", Addison Wesley
4. Matt Weisfeld, "The Object-Oriented Thought Process", Pearson
5. Cox Brad, "Object –Oriented Programming: An Evolutionary Approach", Addison –Wesley

Course Code	**CSL 210**	**Course Title**	**Data Structures with Applications**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			2	**0**	**2**	**3**
Prerequisite (If any)	**Data structures(CSL-102)**	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

1. Ability to design and analyze the applications based on dynamic memory allocation such as linked lists.
2. Ability to apply and relate the concepts of height balanced trees for comparative analysis, and their applications to real world.
3. Ability to incorporate the knowledge of tries and skip lists for different applications.
4. Ability to apply the knowledge of graph data structures for various applications and algorithm design paradigms.

Course Contents:

Module 1:

Applications of lists in polynomial representation, multi-precision arithmetic, Hash-tables, Radix Sort etc. Multi linked structures and an example application like sparse matrices. Implementation of priority queues.

Module 2:

Overview of Binary Search Tree (BST), Height-balanced (AVL) trees, insertion/deletion and rotations. Heaps and heapsort.

Multi-way trees and external sorting - B-trees – insertion and deletion, Introduction to B+ trees with insertion and deletion algorithms. Red-black trees, Splay trees.

Module 3:

Tries, Multi-way tries, Suffix trees, Segment trees. Applications of the above mentioned trees. Introduction to Skip lists, Data structures for disjoint set representation

Module 4 :

Overview and definition of Graph as data structure, Traversals (BFT, DFT, Topological Sort), Data structures for Dijkstra's Shortest Path Algorithm, All-pairs shortest paths, Minimum spanning trees

– Algorithms (Kruskal, Prim) and data structures. Huffman coding. Introduction to network flow problem.

Text Books:

1. Data Structures & Program Design in C: Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.
2. Fundamentals of Data Structures in C : E. Horowitz, S. Sahni, and S. Anderson-Freed, University Press

Reference:

1. Aho, Hopcroft and Ullmann, ―Data Structures and Algorithms,‖ Addison Wesley, 1983.

Course Code:	**CSP 201**	**Course Title:**	**IT Workshop-I**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **0** **0** **4** **2**
Prerequisite (If Any)	**None**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

1. Effectively use the Unix programming environment - shell, file system, scripts, filters, program development tools.
2. Develop good programming style using Python with usage of packages: math, Cmath and functions.
3. Be familiar with writing of real time application programs using the concepts like class, object, inheritance, constructor, tkinter.
4. Use of effective procedures and tools for data analytics using graphical outcomes: Pandas, numpy, openpyxl and matplotlib.
5. Learn to automate tasks for making predictions using machine learning: scikit learn, countplot.

Course Contents:

Module 1:

Introduction to different tools for identification and possibility of errors in C program – gdb, concepts of “core dump”, backtracing using “bt”, using “info” to dump all registers, creating watch-list / watch variables. DDD (Data Display Debugger) – introduction and usage, debugging with ddd (step, step into, step over). Using DevCpp and/or VisualStudio b. Setting compiler options and linker options. Unix tools - Awk, sed, Emacs. Make files and automated builds.

Module 2:

Text editors. Users, files, permissions, and processes on Linux. Introduction to shell: Set and Unset a variable, Displaying – using echo, Using Expr & Test, Getting input – using read, Header files of shell script – using Shabang, Sample Shell script program. Assigning a command to a variable, Storing output

to a variable, Assigning global value – using Export. Command Line Arguments, Conditional & Looping Statement, Functions.

Module 3:
Advanced Commands: SED, Replacing values in a file, STTY, TOP, Sending an email using MAIL, HERE. Scheduler: Scheduling a job – using 'Crontab', 'at' and 'nohup. Shell Programming: Essential systems administration with shell scripting and elementary Python, Version control. Advanced Shell Scripting: Monitoring a file, Handling Shell Script Interrupts, Extracting data from HTML/XML file, Trapping Signals Database Connectivity, Connecting MYSQL to Shell, Running SQL queries from Shell Script.

Module 4:
Bash and Bash Scripting: Common shell programs, Advantages of BASH, Executing commands, Building blocks, developing good scripting, variables, conditionals, loops, finding logged in users. Writing and Debugging Scripts.

Module 5:
Bash Environment: Shell Initialization files, Quoting characters, Shell expansion, Aliases and More options in Bash. Regular Expressions: Meta characters, Extended regular expressions Using GREP, Pattern matching. Python Integration, Testing and Debugging with Software Development Practice.

Text Books:
1. Christopher Negus "Linux Bible", Wiley
2. Steve Parker "Shell Scripting: Expert Recipes for Linux, Bash & more" Wrox
3. Richard Petersen "Linux: The Complete Reference", TMH
4. Robert Collins "Shell Programming and Bash Scripting: Ultimate Beginners Guide Book", CreateSpace

Course Code:	**CSL 212**	**Course Title:**	**Computer Architecture and Organization**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **0** | **0** | **3**
Prerequisite (If Any)	**None**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:
1. Identify the various components for the construction of CPU.
2. Design arithmetic operations for the construction of CPU.
3. Analyze and design control unit for the CPU.
4. Analyze and design memory mapping for the storage requirements of CPU.
5. Analyze and evaluate the performance of CPU through system hardware and pipelining.

Course Contents:

Module 1:
Introduction: Basic functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU - registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study - instruction sets of some common CPUs.

Module 2:

Arithmetic: Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic - integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication - shift-and-add, Booth multiplier, carry save multiplier, etc. Division - non-restoring and restoring techniques, floating point arithmetic.

Module 3:
CPU Design: control unit design: hardwired and micro-programmed design approaches, Case study - design of a simple hypothetical CPU. Performance: CPU performance and its factors, evaluating performance, comparing performance, Amdahl's law, MIPS as a performance measure, Processor Datapath Design and Control Mechanism: building single cycle datapath and control, performance of single cycle implementation, building multicycle datapath and control, comparison with the performance of both the implementations.

Module 4:
Enhancing the performance with Pipelining: Pipelined datapath, Pipelined control, Hazards: Structural, Data and Control/Branch Hazards, Forwarding and Stalling techniques, Branch Prediction, Exceptions

Module 5:
Memory system design: semiconductor memory technologies, memory organization. Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs block size, mapping functions, replacement algorithms, write policy.

Module 6:
Peripheral devices and their characteristics: Input-output subsystems, I/O transfers - program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes - role of interrupts in process state transitions Performance enhancement techniques, Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.

Text Books:
1. Patterson, David A., and John L. Hennessy. Computer organization and design ARM edition: the hardware software interface. Morgan kaufmann, 2016..
2. Parhami, Behrooz. Computer architecture. Oxford University Press, New York, NY, USA, 2005.
3. Baer, Jean-Loup. Microprocessor architecture: from simple pipelines to chip multiprocessors. Cambridge University Press, 2009.

Reference Book:
1. Hamacher, Carl, Zvonko Vranesic, Safwat Zaky, and N. Manjikian. Computer Architecture. Mc Graw Hill, 2002.

4th Semester

<table>
<tr><td>Course Code:</td><td>CSL 205</td><td>Course Title:</td><td>Design and Analysis of Algorithms</td></tr>
<tr><td>Category:</td><td>Core</td><td>Credit Assigned</td><td><table><tr><td>L</td><td>T</td><td>P</td><td>C</td></tr><tr><td>3</td><td>0</td><td>2</td><td>4</td></tr></table></td></tr>
<tr><td>Prerequisite (if Any)</td><td>Advanced Data Structures ,</td><td>Type of Course</td><td colspan="2">Computer Science and Engineering</td></tr>
</table>

	Mathematics		

Course Outcomes:

1. Student will be able to derive the recurrence relations for algorithms and analyze the performance of algorithms using asymptotic notations.
2. Student will be able to perform the amortized analysis and evaluate the cost of various operations on the data structure.
3. Student will be able to analyze and apply various algorithm design paradigms for real world applications. Also, evaluate the performance of algorithm based on various parameters.
4. Student will be able to apply and relate various algorithms to solve the problems based on Graphs.

Course Contents:

Module 1:

Mathematical foundations, summation of arithmetic and geometric series, n, n^2, bounding summations using integration, recurrence relations, and solutions of recurrence relations using technique of characteristic equation and generating functions.

Module 2:

Asymptotic notations of analysis of algorithms, analyzing control structures, worst case and average case analysis, amortized analysis, sorting algorithms such as selection sort, insertion sort, bubble sort, heap sort, lower bound proof, elementary and advanced data structures with operations on them and their time complexity.

Module 3:

Divide and conquer basic strategy, binary search, quick sort, merge sort, Fast Fourier Transform etc. Greedy method - basic strategy, application to job sequencing with deadlines problem, minimum cost spanning trees, single source shortest path etc.

Module 4:

Dynamic Programming basic strategy, multistage graphs, all pairs shortest path, single source shortest paths, optimal binary search trees, traveling salesman problem.

Module 5:

Basic Traversal and Search Techniques, breadth first search and depth first search, connected components. Backtracking basic strategy, 8-Queen's problem, graph colouring, Hamiltonian cycles etc. NP-hard and NP-complete problems, basic concepts, nondeterministic algorithms, NP-hard and NP-complete, Cook's Theorem, decision and optimization problems, polynomial reduction.

Text:

1. Introduction to Algorithms : Cormen T.H. et.al : Prentice Hall of India
2. Computer Algorithms : Horowitz, Sahani, Rajsekharan , Galgotia Publications Pvt.Ltd
3. Fundamentals of Algorithms : Brassard, Bratley , Prentice Hall

Course Code:	**CSL 207**	**Course Title:**	**Operating Systems**
Category:	**Core**	**Credit Assigned**	**L** **T** **P** **C** **3** **0** **2** **4**
Prerequisite (if Any)	**Data Structures (CSL 102)**	**Type of Course**	**Computer Science and Engineering**

Course Outcomes:

Students should be able to

1. Identify the structure and design issues of operating systems.
2. Summarise the concepts of process management and relate the underlying programming constructs.
3. Analyse and evaluate the memory management techniques, I/O management and file systems.
4. Implement general operating system concepts using modern operating systems like Unix and others.

Course Contents:

Module 1:

Introduction, basic h/w support necessary for modern operating systems - Services provided by OS, system programs and system calls - brief discussions of evolution of OS - real time and distributed systems: a brief overview of issues.

Processes and 3 levels of scheduling - process control block and context switch - goals of scheduling and different scheduling algorithms - threads: user-level and kernel level.

Module 2:

Process cooperation and synchronization - mutual exclusion and implementation - semaphores, conditional critical regions and monitors - classical inter - process communication problems - message passing.

Deadlocks and strategies for handling them - protection and security issues - access lists, Capabilities.

Module 3:

Memory management techniques - contiguous and non-contiguous - paging and segmentation - translation look-aside buffers (TLB) and overheads - virtual memory and demand paging- page faults and instruction restart - problems of large address spaces - page replacement algorithms and working sets - miscellaneous issues.

Module 4:

File systems, user interface - disk space management and space allocation strategies - examples from UNIX, DOS, Windows etc - directory structures - disk caching - file system consistency and logs - disk arm scheduling strategies.

Module 5:

Linker and Loader - Concept of static and dynamic relocation, external symbols, design of linker, design of object file for different loading schemes.

Common Object file format - Structure of object file and executable file, section or segment headers, symbol table, concept of storage class, string various, data types line insert, character, arrays structures.

Device Drivers - Device programming, system drivers, non-system drivers, virtual drivers, Incorporation of driver routines, Basic device driver operation, character and block drivers.

Text:

1. Silberchatz & Galvin, “Operating System Concepts”, Addison Wesley
2. Tanenbaum A, “Modern Operating Systems”, PHI 2 nd Ed
3. William Stallings, “Operating Systems”, Pearson Publications

Course Code:	**CSL432**	**Course Title:**	**Human Computer Interaction**

Category:	Elective	Credit Assigned	L	T	P	C
			3	0	0	3
Prerequisite (if Any)	Nil	Type of Course	Computer Science and Engineering			

Course Outcomes:

1. Recognize the requirements of designing the user centered and highly usable software systems.
2. Contribute and apply the advancement of Human-Computer Interaction theory and practice.
3. Evaluate methods, quality factors, and data analysis techniques.
4. Analyze interaction problems from a technical, cognitive, and functional perspective.

Course Contents:

Module-1

Introduction to Interaction Design: User Experience, Process of Interaction Design, Goals of Interaction Design and User Experience, Understanding and Conceptualizing Interaction, Conceptual Models, Interface Metaphors, Interaction Types, Paradigms and Frameworks, Cognitive Aspects, Cognition, Cognitive Framework. Social Interaction, Emotional Interaction.

Module 2

Interfaces: Types of Interfaces, Natural User Interfaces, Data Gathering and Key Issues, Data Recording, Interviews, Questionnaires, Observation, Choosing and Combining Techniques, Data Analysis, Interpretation and Presentation of Qualitative and Quantitative Analysis, Simple Analysis, Tools and Theoretical Frameworks, Presenting the Findings.

Module 3

Process of Interaction Design: Introduction, Establishing Requirements, Data Gathering for Requirements, Task Description, Task Analysis, Design, Prototyping and Construction, Conceptual Design and Physical Design, Using Scenarios, Prototypes in Design.

Module 4

Design Process: Interaction Design Basics, Design Rules, Software Lifecycle, Universal Design and Multimodal Interaction, Design for Diversity, GOMS.

Module 5

Evaluation: Introduction, Goals of Evaluation, Evaluation through Expert Analysis and User Participation, Evaluation Framework, Observing and Testing Users, Case Studies.

TEXT BOOKS:

1. Sharp, H., Rogers, Y., and Preece, J, "Interaction Design: Beyond Human Computer Interaction", Third Edition, John Wiley & Sons, Inc., 2011.
2. Alan Dix, Janet E. Finlay, Gregory D. Abowd and Russell Beale, "Human Computer Interaction", Pearson Education, Third Edition, 2004.

REFERENCE BOOKS:

1. Wilbert O. Galitz, "The Essential Guide to User Interface Design: An Introduction to Gui Design Principles and Techniques", Third Edition, John Wiley Sons, 2002.
2. Benyon, D., Turner, P., and Turner, S, "Designing Interactive Systems: People, Activities, Contexts, and Technologies", Addison-Wesley, 2005.

Course Code	**CSL213**	**Course Title**	**Software Engineering and Game Testing**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Prerequisite (If any)	-	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

1. To look at the large scale software development from a broader perspective, and function in multidisciplinary teams.
2. To apply knowledge gained in the course to practical software development situations in methodical way.
3. To design software systems to meet desired needs with realistic constraints
4. To communicate effectively in software development activities.
5. To get an idea about contemporary issues in Software development and engage in life-long learning, understand professional and ethical responsibility.

Course Contents:

Module 1:

Software Engineering Process & Management : Generic view, Capability Maturity Model, Process models-waterfall, evolutionary, incremental etc., unified process, agile view, project management, metrics estimation, project scheduling, risk management Software engineering Principles and Practice : Communication, planning and modelling practices, system engineering and modelling, business process engineering requirement analysis, system analysis- flow oriented and class oriented modelling using data modelling concepts.

Module 2:

Software Design Engineering - Design Concepts: Abstraction Architecture, pattern modularity, information hiding, design classes, refactoring etc., Design of web application, architectural design, component level design, user interface design.

Module 3:

Software Testing and Quality Management: Testing strategies, testing for object oriented software testing for web applications, validation testing etc. Game Quality Factors, Game Quality Appraisal, Game Standards, User Interface Standards coding Standards, Game Quality Measurements, Quality Plans. Test Phases : Pre-production, Test Kickoffs, Alpha Testing , Beta Testing , Gold Testing, Release Certification, Post-Release Testing

Module 4:

The Game Testing Process: "Black Box" Testing, "White Box" Testing, The Life Cycle of a Build test Cases and Test Suites, On Writing Bugs Well. Testing by the Numbers: Testing Progress, Testing Effectiveness, Tester Performance. Combinatorial Testing: Parameters, Values, defaults, numerations, ranges, boundaries :Constructing Tables, Combinatorial Templates, Combinatorial Test, Generation Combinatorial Economics

Module 5:

Test Flow Diagrams: TFD Elements, TFD Design Activities, A TFD Example, Data Dictionary, data Dictionary Application, data Dictionary Reuse, data Dictionary Example. Cleanroom Testing, Test Trees, Ad Hoc Testing and Gameplay Testing, Ad Hoc Testing. Defect Triggers , Regression Testing and Test Reuse Regression Testing, Capture/Playback Testing Capture/Playback Automation Overview

Text Books:

1. Game Testing: All in One by Charles Schultz and R. Bryant.
2. Software Engineering: A Practitioner's Approach by Roger Pressman ; Tata-McGraw Hill

Reference Books:

1. Software Engineering by Ian Sommerville ; Pearson Ed.
2. Game Development Essentials: Game QA & Testing by Luis Levy and Jeannie Novak

Course Code:	**MAL 201**	**Course Title:**	**Numerical Methods and Probability Theory**
Category:	**Core**	**Credit Assigned**	**L** | **T** | **P** | **C** **3** | **1** | **0** | **4**
Prerequisite (if Any)	**MAL 103, MAL 104**	**Type of Course**	**Basic Science**

Course Outcomes:

The students will be able

1. Apply numerical methods to obtain approximate solutions of nonlinear equations.
2. Compare different iterative methods based on their order of convergence.
3. Choose appropriate numerical methods and apply to solve linear systems of equations and ordinary differential equations.
4. Develop probability density functions of random variables and calculate measures of central tendency.
5. Predict missing data using various distributions.

Course Contents:

Numerical Analysis: Solutions of algebraic and transcendental equations by Iteration method, method of false position, Newton-Raphson method and their convergence.

Solutions of system of linear equations: Gauss elimination method, Gauss Seidal method, LU decomposition method. Newton-Raphson method for system of nonlinear equations. Eigen values and eigen vectors: Power and Jacobi methods.

Numerical solution of ordinary differential equations: Taylor's series method, Euler's modified method, Runge-Kutta method, Adam's Bashforth and Adam's Moulton, Milne's predictor corrector method. Boundary value problems: Shooting method, finite difference methods.

Probability theory: Random variables, discrete and continuous random variable, probability density function; probability distribution function for discrete and continuous random variable joint distributions. Definition of mathematical expectation, functions of random variables, The variance and standard deviations, moment generating function other measures of central tendency and dispersion, Skewness and Kurtosis. Binomial, Geometric distribution, Poisson distribution, Relation between Binomial and Poisson's distribution, Normal distribution, Relation between Binomial and Normal distribution. Random processes, continuous and discrete, determinism, stationarity, ergodicity etc. correlation functions, autocorrelation and cross-correlation, properties and applications of correlation functions.

Text Book:

1. Jain, Iyengar and Jain, Numerical Methods for Engineers and Scientists, Wiley Eastern.
2. Kendall Atkinson, Weimin Han, Elementary Numerical Analysis, Wiley.
3. Gerald and Wheatley Applied Numerical Analysis, Addison-Wesley.
4. Paul L. Meyer, Introductory Probability and Statistical Applications, Addison Wesley.
5. Miller and Freund, Probability and Statistics for Engineers Eastern Economy Edition, PHI.
6. Ross, A First Course in Probability, Pearson Education India.

Reference:

1. S. D. Cante and C. de Boor, Elementary Numerical Analysis, an algorithmic approach, McGraw-Hill.
2. M. R. Spiegel, Theory and problems of Probability and statistics; McGraw-Hill Book Company; 1980.

Course Code	**CSP202**	**Course Title**	**IT Workshop - II**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			0	**0**	**4**	**2**
Prerequisite (If any)	-	**Type of Course**	**Computer Science and Engineering**			

Course Outcome:

1. Understand both the basic and advanced concepts Android OS, gradle, Android Studio.
2. Debug Android Application, Design and develop an application using Database.
3. Develop UI based Mobile Application using Android Studio.
4. Design application, Deploy the application on Google Play.

Course Contents:

Module 1:
Fundamentals: Basic Building blocks – Activities, Services, Broadcast Receivers and Content providers, UI Components – Views and notifications Components for communication –Intents and Intent Filters

Module 2:
Application Structure: AndroidManifest.xml, user-permission – sdk, Resources and R.java, Assets, Layouts and Drawable Resources, Activities and Activity lifecycle.

Module 3:
Emulator-Android Virtual Device: Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS

Module 4:
Basic UI design: Form widgets, Text Fields, Validation of EditText, Layouts, [dip, dp, sip, sp] versus px. Different UI design patterns
Preferences: Shared Preferences, Preferences from xml.
Menu: Option menu, Context menu, menu from xml, menu via code
Intents: Explicit Intents, Implicit intents

Module 5:
UI design: Time and Date, Images and media, Android Adapter and ListView, Composite, Alert Dialogs and Toast, Popup, Fragments, Navigation drawer.
Tabs, Tab Activity Styles & Themes: styles.xml, drawable resources for shapes, gradients (selectors), style attribute in layout file, Applying themes via code and manifest file.
Content Providers: SQLite Programming, SQLite Open Helper, SQLite Database, Cursor, Reading and updating Contacts, Reading bookmarks

Reference Books:

1. Joseph Annuzzi Jr, Lauren Darcey, Shane Condor, "Advanced Android Application Development, Developers Library", Pearson Education, 4th Edition (2015)
2. Lauren Darcey, Shane Condor, "Android, Wireless Application Development", Pearson Education, 3rd Edition.
3. Paul Deitel, Harvey Deitel, Alexander Wald, "Android 6 for programmers, An AppDriven Approach", Pearson Education.
4. Rap Payne, "Beginning App Development with Flutter: Create Cross-Platform Mobile Apps", Apress (2019).

3rd Year Syllabus

5th Semester

Course Code :	**CSL302**	**Course Title :**	**Computer Networks**			
Category :	**CORE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (if Any) :	**None**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:
On completion of the course, students will be able to:
1. Analyse various issues and their solutions at different layers of network architecture
2. Design and develop various networking algorithms
3. Apply networking protocols on a given network to analyze their working
4. Apply networking concepts to build real world networking systems using most important protocols in use today

Course Contents:
Module 1:
Introduction to Computer Networks, Network Architecture: Layering and Protocol, Internet architecture, Implementing Network Software: Application Programming Interface (Socket), Delay x bandwidth product.

Module 2:

Classes of Links, Framing, Error Detection: cyclic redundancy check, Internet checksum algorithm. Reliable transmission: Stop-and-wait, Sliding Window, Ethernet and Multiple Access Networks (802.3), Wi-Fi/802.11

Module 3:
Datagram and virtual circuit switching, Bridges and LAN switches, Basic Internetworking (IP): Internetwork, Global addresses, Datagram forwarding in IP, Subnetting and Classless addressing,

Module 4: Address Translation (ARP), Host Configuration (DHCP), Error Reporting (ICMP), Virtual Networks and tunnels, Distance-vector (RIP), Link-state (OSPF), Routing areas, Interdomain routing (BGP).

Module 5:
Simple demultiplexer (UDP), Reliable byte stream (TCP): End-to-end issues, segment format, Connection establishment and termination, Sliding window, Triggering transmission and Adaptive retransmission, TCP Congestion Control: Additive increase/ Multiplicative decrease, Slow start, Fast retransmission and fast recovery. Resource allocation in TCP

Module 6: Introduction to applications and related Protocols: Electronic mail (SMTP, MIME, IMAP), World Wide Web (HTTP), Name service (DNS).

Text Books:

1. Larry L. Peterson, Bruce S. Davie, "Computers Networks: A systems approach", Morgan Kaufmann, 5th Edition.

Reference Books:

1. Tanenbaum A. S, "Computer Networks", PHI
2. James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition.
3. William Stallings, "Data and Computer Communications", PHI 6th Edition
4. Behrouz A Forouzan, "Data Communication and Networking", 4th Edition
5. Simon Haykin, "Communication Systems", John Wiley 4th Edition
6. Douglas Comer, "Computer Networks and Internets", Addison Wesley 2nd Edition
7. Peterson, Simon, "Computer Networks: A Systems Approach", Pearson Education, Asia

Course Code:	**CSL 431**	**Course Title:**	**Computer Graphics**			
Category :	**ELECTIVE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**0**	**3**
Prerequisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Student will able to understand the structure of modern computer graphics systems and the basic principles of implementing computer graphics primitives
2. Student will be able to analyze and implement key algorithms for modeling and rendering graphical data
3. Student will be able to develop design and problem solving skills with application to computer graphics
4. Student will acquire experience in constructing interactive computer graphics programs

Course Contents:

Module 1:

Introduction of Graphics Systems: Use of Computer graphics, Video Display Devices, Refresh Cathode-Ray Tubes, Raster and Random Scan Displays, Color CRT Monitors, Direct View Storage Tubes, Flat Panel Displays, Three-Dimensional Viewing Devices, Stereoscopic & Virtual Reality Systems, Raster and Random Scan Systems, Different Input and Hard Copy Devices, Graphics Software.

Module 2:

Output Primitives: Points and Lines, Line Drawing Algorithms (DDA & Bresenham's), Circle and Ellipse Generating Algorithms, Conic Sections.

Module 3:

2D Geometric Transformations: Different types of transformations and their matrix representations, Homogeneous Coordinates, Composite Transformations, transformations between Coordinate Systems, Affine transformations, Window-to-Viewport Coordinate transformation, Clipping-Point, Line, Polygon, Curve and Text Clipping.

Module 4:

3D Concepts and Object Representation: Three Dimensional Display Methods, Polygon Surfaces, Curved Lines & Surfaces, Qudric Surfaces, Spline Representations, Cubic Spline interpolation methods, Bezier Curves and Surfaces.

Module 5:

3D Transformations and Viewing: Translation, Rotation, Scaling, Reflection, Shears, Composite Transformations, Projections- Parallel and Perspective, Projection Transformations, Clipping.

Module 6:

Visible Surface Detection Methods: Classification of Visible Surface Detection Algorithms, Back Face Detection, Depth Buffer Method, A-Buffer Method, Scan-Line Method, Depth Sorting Method, BSP-Tree Method & Area Subdivision Method. Illumination Models and Surface Rendering: Light Sources, Basic Illumination Models, Polygon- Rendering Methods.

Text Books:

1. 1. D. Hearn & M.P. Baker - Computer Graphics, 2/e , Pearson Education, New Delhi, 2005

Reference Books:

2. W.M. Newman.- Principle of Interactive Computer Graphics, Mc Graw Hill

Publication, New Delhi, 1995.

3. S. Harrington -Computer Graphics- A Programming Approach, McGraw Hill Publication, New Delhi, 1994.
4. J.D. Foley - A Fundamental of Computer Graphics Addition Wesley, London, 1993.

Course Code:	**CSL 303**	**Course Title:**	**Theory of Computation**

Category :	CORE	Credit Assigned :	L	T	P	C
			3	1	0	4
Prerequisite (if Any) :	NONE	Type of Course :	Computer Science and Engineering			

Course Outcomes:

1. Solve computational problems based on computability and complexity.
2. Design computational models for various grammar and languages.
3. Prove the grammar, language, and automata by using formal mathematical methods.
4. Apply the concepts of theory of computation in developing engineering applications such as compiler design.

Course Contents:

Module 1:

Preliminaries - Sets, operations, relations, transitive closure, countability and diagonalization, induction and proof methods- pigeon-hole principle and simple applications - concept of language - grammars and production rules - Chomsky hierarchy.

Module 2:

Regular grammars, deterministic finite automata - non determinism, conversion to deterministic automata- e-closures, minimization of automata

Module 3:

regular expressions, regular sets, Pump lemma for regular sets- closure properties of regular sets, decision properties for regular sets,

Module 4:

Context - free languages, parse trees and ambiguity, reduction of CFGS, Chomsky and Griebach normal forms, push - down Automata (PDA), non determinism, acceptance by two methods and their equivalence, CFLs and PDAs – Pumping lemma for context free languages, Closure and decision properties of CFLs.

Module 5:

Turing Machines variants, recursively enumerable (r.e.) sets, recursive sets, TM as computer of function, decidability and solvability, Halting Problem, reductions, Post correspondence Problem (PCP) and unsolvability of ambiguity problem of CFGs.

Module 6:

primitive recursive and partial recursive functions Church -Turing thesis - convergence of view points of what "computability" is : Semi formal treatment.

Text Books:

1. Martin John, "Introduction to languages and the theory of computation", TM
2. Hotwani Hopcroft, Ullman, "Introduction to Automata Theory, Languages and computation", Pearson Education

Reference Books:

1. Michael Sipser, "Introduction to the theory of Computation", 3rd edition, Cengage Learning

Course Code	CSL306	Course Title	UI and UX Design			
Category	Core	Credit Assigned	L	T	P	C

			3	**0**	**2**	**4**
Prerequisite (If any)	**NONE**	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

1. Apply the methodology of user experience in the design of UX
2. Develop an user specific UX design for web, mobile and tablets
3. Apply the concept of user interface in the design of UI.
4. Develop web, mobile and tables based application with good UI design
5. Design and develop user centric applications based in the UX and UI design concepts.

Course Contents:

Module 1:

UX introduction: Design thinking, user centered design principles, Role, requirement & ROI, feature prioritization, process model – waterfall, Agile, Scrum

Research – user experience research methods, user persona, scenario creation, empathy mapping and affinity mapping, interviews – user, stakeholder. Conducting survey

Module 2:

Analysis – competitor analysis, user centered analysis, Heuristic analysis, task flow evaluation, attributes, goals, perspectives & Pain points, Google analytics, human computer interface,

Design strategy – intuitive & persuasive designing, design user flows, entry points, effective content strategy, low fidelity sketching, paper prototype, site maps, informational architecture, navigational models, mental models, early usability testing

Module 3:

Sketching, framing & prototype – customer experience framework, low and high fidelity models, workflows, prototype – web, mobile & tablet, interactive prototype, rapid prototype – sprint, error handling, minimum viable product (MVP), tools – Axure, Balsamiq, Invision

Testing – Usability testing, remote usability testing, task grids, feedback analysis, reiterate

Module 4:

Elementor and its powerful plugins. Build a website without a single code. Learn how to build a landing page. Using various plugin to organise and generate leads. Typography and colours in digital design. Composition & Contrast. Create Pages and Add to menu. Build custom header & Footer. Composition: make your website responsive to all screens (table, mobile) Animation in web design. Integrate a Blog onto website. Using SEO in our blog and Website. Connecting the website to google consol and Google Analytics

Module 5:

Basics of User interface design – UI design process, design psychology, web, Mobile & Tablet visual designing, human factors & Ergonomics

Visual Design – storyboards creation, product design, colour theory, layouts, Typography, infographics, iconography, branding design, image editing, web template design, creating UI elements, UI Kits & Image widgets

Google Map Designing, Visual design output & documentation, High impact presentation

Text Books:

1. Joel Marsh, "UX for Beginners", O'Reilly , 2022
2. Jon Yablonski, "Laws of UX using Psychology to Design Better Product & Services" O'Reilly 2021
3. Everett N McKay, UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication, 2013.
4. Don Norman, The Design of Everyday Things: Revised and Expanded Edition, 2013.
5. Steve Krug, Don't Make Me Think: A Common Sense Approach to Web Usability, 2nd ed., 2005
6. Jesse James Garrett, The Elements of User Experience: User-Centered Design for the Web and Beyond, 2nd ed, 2010.

Reference Books:

1. Russ Unger, A Project Guide to UX Design: For user experience designers in the field or in the making, 2nd ed., 2012.
2. Jeff Johnson, Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines, 2014
3. Chris Nodder, Evil by Design: Interaction Design to Lead Us into Temptation, 2013.
4. Jon Yablonski, Laws of UX: Using Psychology to Design Better Products & Services 1st Edition, 2020.
5. Andrew Couldwell, Laying The Foundations: How to Design Websites and Products Systematically, 2020.
6. Jenifer Tidwell, Charles Brewer, Aynne Valencia, "Designing Interface" 3 rd Edition, O'Reilly 2020
7. Steve Schoger, Adam Wathan "Refactoring UI", 2018.
8. Steve Krug, "Don't Make Me Think, Revisited: A Commonsense Approach to Web & Mobile", Third Edition, 2015.
9. https://www.nngroup.com/articles/
10. https://www.interaction-design.org/literature.

Course Code:	**CSL 301**	**Course Title:**	**Database Management System**			
Category :	**CORE**	**Credit Assigned :**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (if Any) :	**NONE**	**Type of Course :**	**Computer Science and Engineering**			

Course Outcomes:

1. Student will be able to design and develop database using ER model with various SQL constraints and apply normalization for consistency in database.
2. Student will be able to write queries using relational algebra, tuple and domain relational calculus, and SQL to retrieve information from database based on data centric applications.
3. Student will be able to analyze and apply the concept of storage management and query processing to fine tune the performance of database at the time of information retrieval.
4. Student will be able to analyze and apply the conception of transaction processing, concurrency control and recovery mechanism in database.

Course Contents:

Module 1:
Database system concepts and Architecture - concept of relational database, Relational data model, Relational algebra, SQL-the relational database standard, ER and EER model.

Module 2:
Database design theory - Functional dependencies and normalization, relational database design algorithms, practical database design and demoralization, Relational constants, programmatic ways for implementing constraints, triggers, Chase algorithm.

Module 3:
Physical database design - Concept of physical and logical hierarchy, storage structures like cluster, index organized table, partitions, various table storage parameters and block storage parameters, concept of index, B-trees, hash index, function index, bitmap index.

Module 4:
Process and memory management in database - Various types of tasks in database, database buffer management, log buffer management code reuse, concept of two tier and N-tier architecture, data dictionary and catalog information database recovery technique. Arier Algorithm for recovery.

Module 5:
Query optimization and performance tuning - Various techniques for query optimization, strong and weak equivalence, cost base optimization, Use of different storage structures in query optimization.

Module 6:
Transaction Processing - Transaction and system concepts, Desirable properties of transaction, Schedules and recoverability, serializability of schedules, concurrency control, lock base protocols and time stamp based protocols, read consistency.

Text Books:
1.Fundamentals of Database Systems : Elmasiri and Navathe, Addisson Wesley, 2000
2.Principles of Database Systems : Ullman , Golgotia Publications 1988

6th Semester

Course Code	**CSL 307**	**Course Title**	**Augmented and Virtual Reality**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Prerequisite (If any)	**Data Structures (CSL 102)**	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

- To apply the concepts and perspective of VR/AR landscape i.e. past, present and future for new AR/VR design
- To apply the fundamentals of computer vision, computer graphics and human-computer interaction techniques related to VR/AR
- To apply the insights to key application areas for VR/AR
- Able to design and implement VR/AR experiences

Course Contents:

Module 1:
Introduction of Virtual Reality (VR): Fundamental Concept and Components of Virtual Reality. Primary Features and Present Development on Virtual Reality.

Multiple Models of Input and Output Interface in VR: Input --Tracker, Sensor, Digital Glove, Movement Capture, Video-based Input, 3D Menus & 3DScanner etc. Output -- Visual /Auditory / Haptic Devices.

Module 2:
Visual Computation in VR: Fundamentals of Computer Graphics. Software and Hardware Technology on Stereoscopic Display. Advanced Techniques in CG: Management of Large Scale Environments & Real Time Rendering.

Module 3:
Interactive Techniques in VR: Body Track, Hand Gesture, 3D Manus, Object Grasp.

Module 4:
Development Tools and Frameworks in VR: Frameworks of Software Development Tools in VR. X3D Standard; Vega, MultiGen, Virtools etc.

Module 5:
Application of VR in Digital Entertainment: VR Technology in Film & TV Production. VR Technology in Physical Exercises and Games. Demonstration of Digital Entertainment by VR.

Module 6:
Augmented and Mixed Reality, Taxonomy, technology and features of augmented reality, difference between AR and VR, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality, wireless displays in educational augmented reality applications, mobile projection interfaces, marker-less tracking for augmented reality, enhancing interactivity in AR environments, evaluating AR systems.

Text Books:

1. D. Schmalstieg and T. Höllerer. Augmented Reality: Principles and Practice. Addison-Wesley, Boston, 2016, ISBN-13 978-0-32-188357-5
2. Steven M. LaValle. Virtual Reality. Cambridge University Press, 2017.
3. Burdea, G. C. and P. Coffet. Virtual Reality Technology, Second Edition.Wiley-IEEE Press, 2003/2006.
4. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, 2013
5. Kelly S. Hale (Editor), Kay M. Stanney. Handbook of Virtual Environments: Design, Implementation, and Applications, Second Edition (Human Factors and Ergonomics), 2014.

Reference Books:

1. Alan Craig, William Sherman and Jeffrey Will, Developing Virtual Reality Applications, Foundations of Effective Design, Morgan Kaufmann, 2009.
2. Jason Jerald. The VR Book: Human-Centered Design for Virtual Reality. Association for Computing Machinery and Morgan & Claypool Publishers, 2015.

Course Code	**CSL308**	**Course Title**	**Computer Vision Techniques**			
Category	**CORE**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-requisite (If any)	**---**	**Type of Course**	**Computer Science & Engineering**			
Course Outcomes: 1. identify basic concepts, models and methods in the field of computer vision						

2. assess which methods to use for solving a given problem, and analyse the accuracy of the methods
3. develop and apply computer vision techniques for solving real world problems
4. choose appropriate image processing methods for image filtering, image restoration, image reconstruction, segmentation, classification and representation

Course Contents:

Module 1:
Digital Image Formation and low-level processing: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Module 2:
Depth estimation and Multi-camera views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration.

Module 3:
Feature Extraction: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.
Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Module 4:
Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Module 5:
Shape from X: Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

Text Books:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.

Reference Books:

1. Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006.
2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
3. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.
4. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.

Course Code	**CSL 309**	**Course Title**	**GPU Computing**			
Category	**Elective**	**Credit Assigned**	**L**	**T**	**P**	**C**
			2	**0**	**2**	**3**
Pre-requisite (If any)	-	**Type of Course**	**Computer Science and Engineering**			
Course Outcomes:						
Module 1						

GPU architectures - Streaming Multi Processors, Cache Hierarchy, The Graphics Pipeline, Architectural differences between CPU and GPU

Module 2

Introduction to CUDA programming, Thread organization, Multi-dimensional mapping of dataspace, Synchronization Warp Scheduling, Divergence Memory Access Coalescing, Matrix-matrix multiplication using CUDA with and without memory coalescing

Module 3

GPU memory: Global, shared, and texture memory

GPU Optimization techniques and examples: optimizing Reduction Kernels, Kernel Fusion, Thread and Block Coarsening, Optimization using shared memory

Page-Locked & Zero-Copy Host Memory

Module 4

OpenCL basics, comparative illustration between OpenCL and CUDA

CPU GPU Program Partitioning Application Design: Efficient Neural Network Training/Inferencing

Multi GPU environment: load distribution, communication strategies, and efficient utilization of communication bandwidth

Text Books:

1. Programming Massively Parallel Processors: A Hands-On Approach, 2nd Edition, David Kirk and Wen-mei Hwu, Publisher: Morgan Kaufman, 2012, ISBN: 9780124159921.

Reference material:

CUDA: docs.nvidia.com/cuda

Course Code	**CSL 310**	**Course Title**	**2D and 3D Game Development**			
Category	**Core**	**Credit Assigned**	**L**	**T**	**P**	**C**
			3	**0**	**2**	**4**
Pre-requisite (If any)	-	**Type of Course**	**Computer Science and Engineering**			

Course Outcomes:

On Successful completion of the course, student will be able to:

1. Develop programming skills using software environment of a game engine and its scripting language
2. Apply the concepts of Object-Oriented principle for efficient game code development.
3. Design and implement new features and game mechanics for different games.
4. Create and design game ready 2D and 3D models for different game scenarios.
5. Create realistic scenes and environments for 2D and 3D games, design, implement and deploy games.

Course Contents:

Module 1: Introduction

Introduction to 2D & 3D games – 2D game, 3D game, different game architectures, hardware requirements.

Motivation; Types of games, Different aspects of game design; Different components in a game; Game engines; Geometric primitives, 2D and 3D linear transforms, Homogeneous matrices; Examples of games. Hardware capabilities for gaming.

Module 2: Sprites and animation

Introduction to game design – platforms, input devices, game genres. Game design documentation – what is game design document? sections of game design document.

Different image formats; Polygon file formats; Creating sprites; Rigging; Animations using sprite-sheets; Animations using keyframes; Animation controllers.

Module 3: Designing Levels

Scenes; Tiles, visual continuity in tiles; Adding objects to scene; automatic objects creation; Prefabs; Lighting, RGB space, transparency, texture mapping; Collectibles; Navigation and pathfinding. Creating the environment.

Module 4: Interaction with World

Physics engines; Gravity simulation; Rigid body interaction;
Collision detection and reactions - Collision detection between point, circles, axis, line/point, line/circle, etc., finding out who hits what, Collision reactions / corrections, Common Issues with Collision Detection. Instantiation and Rigid bodies.

Module 5: User interface and Media

Interface; Layout; Menu system; Visual components; Event system; Skins. Audio assets; Different audio formats; Audio mixing. Creating resources – graphics, sound and music, fonts, shaders. Developing game mechanics – general purpose, 2D & 3D platforms, Rhythm Games, Match-x Games, cutscenes.

Module 6: Case studies

Particle system, designing menu, animation, building shapes, case studies.

Text Books:

1. 2D Game Development: From Zero To Hero. Penaz, et al. Internet Game Community; eBook, 2014.
2. Unity 2D game development. Calabrese, Dave. Packt Publishing, 2014.
3. Unity 3.x game development essentials: game development with C# and Javascript. Goldstone, W. Packt publishing, 2nd edition, 2011.

Reference Books:

1. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C. Bond, Jeremy Gibson. Addison-Wesley professional, 2014.
2. Getting Started with Unity 2018: A Beginner's Guide to 2D and 3D game development with Unity. Lavieri, Edward. Packt Publishing Ltd, 2018.
3. Unity Game Development Cookbook: Essentials for Every Game, Paris Buttfield-Addison et al., 1st Edition, O'Reilly Media, 2019.
4. Practical Game Development with Unity and Blender. Thorn, Alan. Delmar Learning, 2014.
5. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C. Bond, Jeremy Gibson. Addison-Wesley professional, 2014.