Scheme & Syllabus of

Bachelor of Technology

Software Engineering

Batch 2019 onwards

By

Department of Academics

IK Gujral Punjab Technical University

Bachelor of Technology in Software Engg.

It is a Graduate (UG) Programme of 4 years duration (8 semesters)

Courses & Examination

First Semester

Course Code	Course Type	Course Title	Load Allocations			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
SE 1110	Engineering Science Course	Programming for Engineers 1	3	0	0	40	60	100	2
EP 1150	Basic Science Course	Physics for Engineers 1	3	0	0	40	60	100	2
EP 1700	Engineering Science Course	Engineering Mechanics 1	3	1	0	40	60	100	3
EP 1990	Engineering Science Course	Introduction to Engineering Measurements	3	1	0	40	60	100	2
MA 1130	Basic Science Course	Enriched Calculus 1	3	2	0	40	60	100	3
MA 1300	Basic Science Course	Linear Algebra for Engineers	3	2	0	40	60	100	3
SE 1110P	Engineering Science Course	Programming for Engineers 1 Lab	0	0	2	30	20	50	1
EP 1150P	Basic Science Course	Physics for Engineers 1 Lab	0	0	2	30	20	50	1
Total			**18**	**6**	**4**	**420**	**280**	**700**	**17**

Second Semester

Course Code	Course Type	Course Title	Load Allocations			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
SE 1210	Engineering Science Course	Programming for Engineers 2	3	0	0	40	60	100	2
EP 1250	Basic Science Course	Physics for Engineers 2	3	0	0	40	60	100	2
MA 1230	Basic Science Course	Enriched Calculus 2	3	2	0	40	60	100	3
MA 1700	Engineering Science Course	Discrete Mathematics	3	1	0	40	60	100	3
CH 1520	Basic Science Course	Principles of Chemistry	3	0	3	40	60	100	3
DR 1520	Engineering Science Course	Engineering Graphics	2	0	3	40	60	100	3
SE 1210P	Engineering Science Course	Programming for Engineers 2 Lab	0	0	2	30	20	50	1
EP 1250P	Basic Science Course	Physics for Engineers 2 Lab	0	0	3	30	20	50	1
Total			**17**	**3**	**11**	**420**	**280**	**700**	**18**

Third Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTES301-18	Engineering Science Course	Digital Electronics	3	0	0	40	60	100	3
SE301-19	Professional Core Courses	Data structure & Algorithms	3	1	0	40	60	100	3
SE302-19	Professional Core Courses	Computer Networks	3	1	0	40	60	100	3
SE303-19	Professional Core Courses	Software Engineering	3	0	0	40	60	100	3
HSMC 101/102-18	Humanities & Social Sciences Including Management \Courses	Foundation Course in Humanities (Development of Societies/Philosophy)	2	1	0	40	60	100	3
BTES302-18	Engineering Science Course	Digital Electronics Lab	0	0	2	30	20	50	1
SE304-19	Professional Core Courses	Data structure & Algorithms Lab	0	0	4	30	20	50	2
SE305-19	Professional Core Courses	Computer Networks lab.	0	0	2	30	20	50	1
SE306-19	Professional Core Courses	Software Engineering lab.	0	0	2	30	20	50	1
SE307-19	Professional Core Courses	IT Workshop*	0	0	2	30	20	50	1
		Summer Institutional Training	0	0	0	60	40	100	Satisfactory/Un satisfactory
Total			**14**	**3**	**12**	**410**	**440**	**850**	**21**

*Syllabus to be decided by respective institute internally. It may include latest technologies.

Fourth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
SE401-19	Professional Core Courses	Database Management system	3	0	0	40	60	100	3
SE402-19	Engineering Science Course	Computer Organization & Architecture	3	1	0	40	60	100	3
SE403-19	Professional Core Courses	Operating Systems	3	1	0	40	60	100	3
SE404-19	Professional Core Courses	Design & Analysis of Algorithms	3	1	0	40	60	100	3
HSMC 122-18	Humanities & Social Sciences including Management Courses	Universal Human Values 2	2	1	0	40	60	100	3
SE405-19	Engineering Science Course	Database Management system Lab	0	0	4	30	20	50	2
SE406-19	Engineering Science Course	Computer Organization & Architecture Lab	0	0	2	30	20	50	1
SE407-19	Professional Core Courses	Operating Systems Lab	0	0	2	30	20	50	1
SE408-19	Professional Core Courses	Design & Analysis of Algorithms Lab	0	0	4	30	20	50	2
Total			**14**	**4**	**12**	**320**	**380**	**700**	**21**

There will be 4-6 weeks summer industrial training after 4th sem.

Fifth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTES 501-18	Engineering Science	Enterprise Resource Planning	3	0	0	40	60	100	3
SE501-19	Professional Core Courses	Agile Software Development	3	0	0	40	60	100	3
BTCS 502-18	Professional Core Courses	Formal Language & Automata Theory	3	0	0	40	60	100	3
SE502-19	Professional Core Courses	Software Testing and Quality Assurance	3	0	0	40	60	100	3
SE XXX-18	Professional Elective	Elective-I	3	0	0	40	60	100	3
EVS101-18	Mandatory Courses	Environmental Sciences	3	-	-	100	-	100	S/US
MC	Mandatory Courses	Constitution of India/ Essence of Indian Traditional Knowledge	2	-	-	100	-	100	S/US
SE503-19	Professional Core Courses	Agile Software Development Lab	0	0	4	30	20	50	2
SE 506-18	Professional Core Courses	Software Testing and Quality Assurance Lab	0	0	2	30	20	50	1
BTCS XXX-18	Professional Elective	Elective-I Lab	0	0	2	30	20	50	1
	Professional Training	Industrial *Training	-	-	-	60	40	100	S/US
Total			20	0	8	550	400	950	19

* 4-6 weeks industrial training undertaken after 4th semester in summer vacations.

Sixth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 601-18	Professional Core Courses	Compiler Design	3	0	0	40	60	100	3
BTCS 602-18	Professional Core Courses	Artificial Intelligence	3	0	0	40	60	100	3
BTCS UUU	Professional Elective Courses	Elective-II	3	0	0	40	60	100	3
BTCS YYY	Professional Elective Courses	Elective-III	3	0	0	40	60	100	3
BTOE ***	Open Elective Courses	Open Elective-I	3	0	0	40	60	100	3
BTCS 603-18	Project	Project-1	0	0	6	60	40	100	3
BTCS 604-18	Professional Core Courses	Compiler Design Lab	0	0	2	30	20	50	1
BTCS 605-18	Professional Core Courses	Artificial Intelligence Lab	0	0	2	30	20	50	1
BTCS UUU	Professional Elective Courses	Elective-II lab	0	0	2	30	20	50	1
BTCS YYY	Professional Elective Courses	Elective-III lab	0	0	2	30	20	50	1
Total			**15**	**0**	**14**	**380**	**420**	**800**	**22**

Seventh Semester / Eighth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 701-18	Professional Core Courses	Network Security and Cryptography	3	0	0	40	60	100	3
BTCS 702-18	Professional Core Courses	Data Mining and Data Warehousing	3	0	0	40	60	100	3
BTOE ***	Open Elective Courses	Open Elective-II	3	0	0	40	60	100	3
BTCS ZZZ	Professional Elective	Elective- IV	3	0	0	40	60	100	3
BTCS TTT	Professional Elective Courses	Elective-V	3	0	0	40	60	100	3
BTCS 703-18	Project	Project-II	0	0	12	120	80	200	6
BTCS ZZZ	Professional Elective	Elective- IV lab	0	0	2	30	20	50	1
BTCS TT	Professional Elective	Elective- V lab	0	0	2	30	20	50	1
Total			**15**	**0**	**14**	**380**	**420**	**800**	**23**

Seventh Semester / Eighth Semester

Course Code	Course Title	Marks Distribution		Total Marks	Credits
		Internal	External		
BTCS 801-18	Semester Training	300	200	500	16

LIST OF ELECTIVES

BTCS XXX-18: Elective-I

BTCS 510-18	Programming in Python
BTCS 513-18	Programming in Python Lab
BTCS 515-18	Computer Graphics
BTCS 518-18	Computer Graphics lab
BTCS 520-18	Web Technologies
BTCS 522-18	Web Technologies lab

BTCS UUU: Elective-II

SE 601-19	Object Oriented Analysis & Design
SE 602-19	Object Oriented Analysis & Design lab
BTCS 614-18	Software Project Management
BTCS 615-18	Software Project Management lab
BTCS 612-18	Cloud Computing
BTCS 613-18	Cloud Computing Lab

BTCS YYY: Elective-III

SE 603-19	Advanced Software Engineering
SE 604-19	Advanced Software Engineering lab
SE 605-19	Service Oriented Architecture
SE 606-19	Service Oriented Architecture lab
SE 607-19	Big Data Analytics
SE 608-19	Big Data Analytics lab

BTCS ZZZ: Elective-IV

BTCS618-18	Machine learning
BTCS 619-18	Machine Learning lab
BTCS 620-18	Mobile Application Development
BTCS 621-18	Mobile Application Development lab
BTCS 704-18	Deep Learning
BTCS 705-18	Deep Learning Lab

BTCS TTT: Elective-V

SE 701-19	Software Metrics
SE 702-19	Software Metrics lab
SE 703-19	Software Verification and Validation Testing
SE 704-19	Software Verification and Validation Testing lab
SE 705-19	Secure Software Development
SE 706-19	Secure Software Development lab

LIST OF COURSES FOR HONOURS DEGREE

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS H01-18	Professional Elective Courses	Graph Theory	3	0	0	40	60	100	3
BTCS H02-18	Professional Elective Courses	Computer Vision	3	0	0	40	60	100	3
BTCS 611-18	Professional Elective Courses	Embedded Systems	3	0	0	40	60	100	3
BTCS H03-18	Professional Elective Courses	Software Project Management	3	0	0	40	60	100	3
BTCS H04-18	Professional Elective Courses	Cryptography & Network Security	3	0	0	40	60	100	3
BTCS H05-18	Professional Elective Courses	Internet-of-Things	3	0	0	40	60	100	3
BTCS 804-18	Professional Elective Courses	Data Analytics	3	0	0	40	60	100	3
BTCS 608-18	Professional Elective Courses	Machine Learning	3	0	0	40	60	100	3
BTCS H06-18	Professional Elective Courses	ICT in Agriculture and Rural Development	3	0	0	40	60	100	3
BTCS H07-18	Professional Elective Courses	Computational Technologies for Smart Cities	3	0	0	40	60	100	3
BTCS H08-18	Professional Elective Courses	Computer Forensics	3	0	0	40	60	100	3

In order to have an Honours degree, a student choose 18-20 credits from the following courses in addition.

Semester-1st

SE 1110	Programming for Engineers 1	3L:0T:2P	3T+1P Credits

Calendar Description: Students are introduced to the concepts of computer programming with specific emphasis on engineering problems and applications. Students learn computer programming as a part of engineering process. Students conceptualize the programming approach in line with engineering profession by following design, implement and testing using specifications. Students explore C++ programming basics, statements, syntax, control structures, functions, and types of arrays.

Prerequisites: Admission to either Electrical, Computer or Software Engineering Program Or Engineering Program Advisor's permission.

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

1. Understand the design, coding and debugging of computer program.
2. Translate the specification into a computer program to solve small engineering problems.
3. Describe the fundamental programming constructs and use them to develop program.
4. Apply the fundamental principles, concepts and constructs of modern computer programming to solve small engineering problems.
5. Demonstrate the control structures, selections and execution flow.
6. Write functions to demonstrate the knowledge of modular programming.

Texts/Materials

B1: Y. Daniel Liang, Introduction to Programming with C++, 3/E, Prentice Hall, 2012, ISBN-10: 0133252817.

Course Topics	**Textbook Mapping**	
	Chapter #	**Book**
Introduction to Computers, Programming, and C++	1	B1
Elementary Programming	2	B1
Selections	3	B1
Mathematical Functions, Characters, and Strings	4	B1
Loops	5	B1
Functions	6	B1
Single-Dimensional Arrays and C-Strings	7	B1
Multidimensional Arrays	8	B1

EP 1150	**Physics for Engineers 1**	**3L:0T:2P**	**3T+1P Credits**

Calendar Description: This course is similar to PHYS 1150: Mechanics and Waves except that Engineering students do not complete the laboratory portion. Students are introduced to and apply calculus to physical concepts. Topics include mechanics, simple harmonic motion, mechanical waves, sound, wave optics and geometric optics.

Prerequisites: Admission to the Engineering Transfer Program.

Educational Objectives/Outcomes:

After completing this course the students will have the ability to:

1. Explain and write about the physical concepts for waves, wave optics, geometric optics, kinematics, and dynamics
2. Apply the physical concepts to complex multiple unknown problems by solving such problems using calculus

Texts/Materials

B1: Serway, R.; Jewett, J.; Physics for Scientists and Engineers, Brooks Cole; 9 edition, 2013, ISBN-13: 978-1133947271.

Course Topics	**Textbook Mapping**	
	Chapter	**Book**
Optics, Electromagnetic Waves, Plane EM Waves, EM Spectrum	34	B1
Geometric Optics, The Ray Approximation, Reflection and Refraction, Dispersion and Prisms, Total Internal Reflection, Optics - Spherical Mirrors, Refraction, Lenses	35, 36	B1
Wave Optics, Conditions for Interference, Young’s Double-slit Experiment, Intensity Distribution, Interference in Thin Films	37	B1
Kinematics and Vectors, Review of Motion in One-Dimension, Displacement, Velocity, Speed, Acceleration, Kinematic Equations, Freely Falling Bodies	2	B1
Vectors Review, Properties of Vectors, Components and Unit Vectors	3	B1

Mechanics (with Calculus), Two-dimensional Motion, Displacement, Velocity and Position Vectors, 2-D Motion with Constant Acceleration, Uniform Circular Motion	4	B1
Force and Motion, Concept of Force, Newton’s Laws, Friction	5	B1
Work and Energy, The Scalar (Dot) Product, Work Done by Constant & Varying Forces, Kinetic Energy and the Work- Energy Theorem, Potential Energy, Conservative and Non- conservative Forces, Conservation of Energy, Change in Energy when the Force is Non-conservative, Power	7, 8	B1
Momentum and Collisions (covered in seminar)	9	B1
Static Equilibrium and Elasticity (covered in seminar)	12	B1
Wave Phenomena, Oscillatory Motion, Simple Harmonic Motion, Mass Attached to a Spring, Energy of the Simple Harmonic Oscillator	15	B1
Wave Motion, Introduction, Types of Waves, One-D Travelling Waves, Speed of Waves on Strings, Reflection and Transmission, Sinusoidal Waves, Energy Transmitted by Waves on Strings	16	B1
Sound Waves, Speed of Sound Waves, Periodic Sound Waves, Intensity of Periodic Sound Waves, The Doppler Effect	17	B1
Superposition and Standing Waves, Superposition and Interference of Sinusoidal Waves, Standing Waves in Strings and Air Columns	18	B1

EP 1700	Engineering Mechanics 1	3L:1T:0P	3 Credits

Calendar Description: Study and apply the principles required to solve engineering mechanics problems. Students explore statics, including equilibrium of particles, equilibrium of rigid bodies, trusses and dry frictions. Further topics of study in dynamics include rectilinear motion, curvilinear motion, Newton's second law, equations of motion in different coordinates, impulse, momentum, impact, work and energy.

Prerequisites: EPHY 1150 or PHYS 1150, MATH 1130 or MATH 1140 or MATH 1300

or MATH 1540

Educational Objectives/Outcomes:

After completing this course the students will have the ability to:

1. Solve statical equilibrium problems for particles subjected to concurrent forces.
2. Solve statical equilibrium problems for rigid bodies subjected to a system of forces and moments.
3. Solve problems involving dry friction.
4. Solve kinematic problems for particles and systems of particles in motion.
5. Use Newton's 3rd Law to solve kinetics problems for particles using rectangular, normal- tangential and polar/cylindrical coordinate systems.
6. Solve kinetics problems for particles and systems of particles using work and energy.
7. Solve kinetics problems for particles and systems of particles using impulse and momentum.

Texts/Materials

B1: Hibbeler, R.C.; Engineering Mechanics, Statics and Mechanics, Prentice Hall, 2015, ISBN-13: 978-0133915426.

M1: Engineering Design Notepaper (Available at the Bookstore), Scientific Calculator.

Course Topics	Textbook Mapping	
	Chapter	Book
Introduction, Fundamental Concepts, Units of Measurement, The International System of Units, Numerical Calculations, General Procedure for Analysis	1.1 to 1.6	B1

Scalars and Vectors, Vector Operations, Vector Addition of Forces, Addition of a System of Coplanar Forces, Cartesian Vectors	2.1 to 2.5	B1
Addition of Cartesian Vectors, Position Vectors, Force Vector Directed Along a Line, Dot Product	2.6 to 2.9	B1
Condition for the Equilibrium of a Particle, The Free-Body Diagram, Coplanar Force Systems	3.1 to 3.3	B1
Three-Dimensional Force Systems	3.4	B1
Moment of a Force – Scalar Formulation, Cross Product Moment of a Force – Vector Formulation, Principle of Moments, Moment of a Force About a Specified Axis	4.1 to 4.5	B1
Moment of a Couple, Simplification of a Force and Couple System, Further Simplification of a Force and Couple	4.6 to 4.8	B1
Reduction of a Simple Distributed Loading	4.9	B1
Conditions for Rigid-Body Equilibrium	5.1	B1
Free-Body Diagrams, Equations of Equilibrium	5.2 to 5.3	B1
Two- and Three-Force Members,	5.4	B1
Free-Body Diagrams, Equations of Equilibrium. Constraints and Statical Determinacy	to 5.7	
Simple Trusses, The Method of Joints	6.1 to 6.2	B1
Zero-Force Members, The Method of Sections	6.3 to 6.4	B1
Characteristics of Dry Friction, Problems Involving Dry Friction, Wedges	8.1 to 8.3	B1
Introduction, Rectilinear Kinematics: Continuous Motion Rectilinear Kinematics: Erratic Motion, General Curvilinear Motion, Curvilinear Motion: Rectangular Components, Motion of a Projectile	12.1 to 12.6	B1
Curvilinear Motion: Normal and Tangential Components Curvilinear Motion: Cylindrical Components, Absolute Dependent Motion Analysis of Two Particles, Relative- Motion of Two Particles Using Translating Axes	12.7 to 12.10	B1
Newton's Second Law of Motion, The Equation of Motion The Equation of Motion for a System of Particles, Equations of	13.1 to 13.5	B1

Motion: Rectangular Coordinates, Equations of Motion: Normal and Tangential Coordinates		
Equations of Motion: Cylindrical Coordinates	13.6	B1
The Work of A Force, Principle of Work and Energy Principle of Work and Energy for a System of Particles, Power and Efficiency	14.1 to 14.4	B1
Conservative Forces and Potential Energy, Conservation of Energy	14.5 to 14.6	B1
Principle of Linear Impulse and Momentum, Principle of Linear Impulse and Momentum for a System of Particles, Conservation of Linear Momentum for a System of Particles	15.1 to 15.2	B1
Impact, Angular Momentum, Relation Between Moment of a Force and Angular Momentum	15.3 to 15.7	B1

EP 1990	Introduction to Engineering Measurements	3L:1T:0P	3 Credits

Calendar Description: Students are introduced to the measurement and control of physical quantities of interest in engineering and scientific applications. Issues and methods relevant to the real-time measurement and control of parameters such as force, displacement, acceleration, temperature, level, pressure, and flow are considered. Students apply the principles developed in the course during seminars, and in discussions of case studies that are relevant to various engineering or scientific disciplines.

Prerequisites: MATH 1130 or 1140, and EPHY 1150 or PHYS 1150, or permission of the instructor.

Educational Objectives/Outcomes:

After completing this course the students will have the ability to:

1. Understand the operating principles of various sensors and instruments used to measure and control physical quantities in engineering applications.
2. Understand the linearity, gain, range, quantization error, and speed of measurement systems.
3. Understand the principles of real- time data acquisition, and measurement systems software development.
4. Use data- acquisition hardware and software to interface measurement sensors, and create software to capture, analyze, and display, and process measured quantities.
5. Assemble and apply elementary sensor signal- conditioning circuits.
6. Use electronic test equipment to test and configure electronic measurement circuits.
7. Apply curve fitting methods for the application of non- linear sensors in measurement.
8. Familiar with case studies and engineering applications involving the measurement of temperature, strain, force, pressure, angular and linear position and speed, and flow.

Texts/Materials

Reference materials will be provided via the course website.

Course Topics
Electrical Measurement fundamentals, Current, voltage, power, resistance, capacitance, inductance, Multimeters, function generators, oscilloscopes, Ohm's and Kirchhoff's laws, Electro-mechanical energy conversion principles, Rotating DC machine principles and measurements
Data Acquisition Principles, Introduction to data acquisition hardware, Sensors and signal conditioning, Linearity, gain, span, sampling rate, and quantization errors, Non- linear sensors, Introduction to data

acquisition software applications.
Sensing Technologies and their Applications, Temperature, Strain, Force, Torque, and Pressure, Flow and Level, Displacement, Position, and Proximity
Principles of Surveying, Coordinate Systems; Electronic surveying transits; Electronic distance measurements; Leveling and positioning a transit; Measuring horizontal angles; Measuring vertical angles; Triangulation with controlled points;

MA 1130	Enriched Calculus 1	3L:2T:0P	3 Credits

Course Description: This course includes an in-depth study of single-variable differential calculus and its applications, and provides a strong foundation for further study in mathematical disciplines. This is a required course for students in the Engineering Transfer program.

Prerequisites: "A" in Pre-calculus Math 12 or equivalent (British Columbia graduates of 2013 onwards) or "A" in Principles of Math 12 or equivalent (British Columbia graduates prior to 2013) or admission to the Engineering program Note: Students who already have credit for MATH 1140, MATH 1150, or MATH 1170 may not take MATH 1130 for further credit.

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

1. Understand the mathematical concept of a limit at a basic level.
2. Understand the derivative as a limit describing instantaneous rates of change.
3. Display reasonable proficiency in working with elementary mathematical functions and their derivatives.
4. Interpret and apply derivatives in applications.
5. Use the computer mathematics system MAPLE when appropriate.

Texts/Materials

B1: W. Briggs, L. Cochran, B. Gillett, Calculus: Early Transcendentals, Single Variable, 2nd Edition, Pearson Education, 2015.

Course Topics	**Textbook Mapping**	
	Chapter	**Book**
Functions, domains, ranges, and intervals, graphs, combinations of functions models	1	B1
Tangent lines and slope predictors, limits, techniques for computing limits, infinite limits, limits at infinity, continuity	2	B1
The derivative and rates of change, basic differentiation rules, power rule, product and quotient rules, derivative of trigonometric functions,	3	B1

derivatives as rates of change, the chain rule, implicit differentiation, derivatives of exponential and logarithmic functions, derivatives of inverse trig functions, related rates, maxima and minima on closed intervals, applied optimization problems		
Maxima and minima, increasing/decreasing functions and the mean value theorem, first derivative test, curve sketching, second derivative test and concavity, graphing functions, applied optimization problems, linear approximation, L'Hopital's rule, antiderivatives	4	B1

MA 1300	**Linear Algebra for Engineers**	**3L:2T:0P**	**3 Credits**

Course Description: This course is designed for students in the first year Engineering Transfer program and covers vectors in n- dimensional space, linear transformations, matrices, Jordan-Gauss elimination method, eigenvalues and eigenvectors and their application to engineering problems. A computer lab component will be used to explore applications.

Prerequisites: Admission to the Engineering program.

Course Vectoring: (3, 2, 0)

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

1. Understand the properties of the vector spaces,
2. Understand the relationship between a linear transformation and the associated matrix,
3. Find eigenvalues and eigenvectors and apply these concepts to solving differential equations.

Texts/Materials

B1: Lay, D., Lay, S., McDonald, J., Linear Algebra and its Applications, Pearson, Fifth Edition (2016).

Course Topics
Linear Equations in Linear Algebra: Systems of Linear Equations, Row Reduction and Echelon Forms, Vector Equations, The Matrix Equation $AA\mathbf{xx}=\mathbf{bb}$, Solution Sets of Linear Systems, Applications of Linear Systems, Linear Independence, Introduction to Linear Transformation, The Matrix of a Linear Transformation
Matrix Algebra: Matrix Operations, The Inverse of a Matrix, Characterization of Invertible Matrices, Subspaces of $\mathbb{R}nn$, Dimension and Rank
Determinants: Properties of Determinants, Cramer’s Rule, Volume and Linear Transformations
Eigenvalues and Eigenvectors: Eigenvalues and Eigenvectors, The Characteristic Equation, Similarity, Diagonalization, complex numbers, Discrete Dynamical Systems, Applications Differential Equations
Orthogonality: Orthogonality, Orthogonal Complement, Orthogonal Projection, The Gram-Schmidt Process

Semester-2nd

SE 1210	**Programming for Engineers-2**	**3L:0T:2P**	**3T+1P Credits**

Calendar Description: Students are introduced to the concepts of object-oriented programming in designing, implementing and testing engineering problems. Students learn the principles of inheritance and polymorphism in designing of methods and classes in object-oriented approach. Students explore the techniques of reading and writing data to file, exceptional handling, pointers, and dynamic memory management, vectors, stacks and recursion.

Prerequisites: A minimum of grade “C” or better in SENG 1110.

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

1. Identify classes, objects, members of a class and the relationships among to model a problem specification using C++.
2. Describe the principles of inheritance and polymorphism and demonstrate their relation to the design of methods and classes in C++.
3. Use pointers and describe the rationale in small engineering applications.
4. Demonstrate the use of reading and writing data to file.
5. Discuss some conceptual use of vectors, stacks and recursion.
6. Understand error-handling techniques using exception-handling techniques.

Texts/Materials

B1: Y. Daniel Liang, Introduction to Programming with C++, 3/E, Prentice Hall, 2012, ISBN-10: 0133252817.

Course Topics	**Textbook Mapping**	
	Chapter #	**Book**
Objects and Classes	9	B1
Object-Oriented Thinking	10	B1
Pointers and Dynamic Memory Management	11	B1
Templates, Vectors, and Stacks	12	B1
File Input and Output	13	B1
Operator Overloading	14	B1

Inheritance and Polymorphism	15	B1
Exception Handling	16	B1
Recursion	17	B1

EP 1250	**Physics for Engineers 2**	**3L:0T:3P**	**3T+1P Credits**

Calendar Description: This course is a continuation of EPHY 1150. Topics include thermodynamics, kinetic theory of gases, electricity and magnetism.

Prerequisites: A minimum of grade “C” in EPHY 1150.

Educational Objectives/Outcomes:

After completing this course the students will have the ability to:

1. Solve problems in electrostatics involving electric forces and electric fields due to point charges by means of vector additions and Coulomb’s law.
2. Calculate electric potentials and potential differences for point charges and uniformly charged conductors.
3. Solve kinematic problems involving point charges moving within a uniform electric/magnetic field.
4. Calculate magnetic forces and torques due to current-carrying conductors inside a uniform magnetic field.
5. Determine magnetic fields around current-carrying conductors by means of vector additions and Biot-Savart law.
6. Use Ampere’s Law to determine the magnetic fields around current-carrying conductors and inside an ideal solenoid.
7. Explain the concepts of magnetic flux and eddy current, and apply Faraday’s and Lenz’s Laws to solve problems in electromagnetic induction and the concepts of thermal equilibrium and absolute temperature, and solve problems involving the thermal properties of solids, liquids and ideal gases.
8. Apply the Laws of Thermodynamics and explain them.

Texts/Materials

B1: R.A. Serway and J.W. Jewett, Physics for Scientists and Engineers, Thomson Learning Inc., 8th ed. (2010) or 9th ed. (2014)

Course Topics	**Textbook Mapping**	
	Chapter	**Book**
Electric Fields: Properties of Charge; Insulators and Conductors; Coulomb’s Law; Electric Fields; Electric Field Lines; Motion of a Charged Particle in a Uniform	23	B1

Electric Field.		
Gauss's Law: Electric Flux; Gauss's Law and Applications.	24	B1
Electric Potential: Potential Difference and Electric Potential; Potential Differences in Uniform Electric Fields; Electric Potential and Potential Energy Due to Point Charges; Obtaining Electric Fields from Electric Potentials; Potential of a Charged Conductor.	25	B1
Magnetic Fields: The Magnetic Field and Vector Products; Motion of a Charged Particle in a Uniform Magnetic Field; Magnetic Force on a Current-Carrying Conductor; Torque on a Current Loop in a Uniform Magnetic Field.	29	B1
Sources of the Magnetic Field: The Biot-Savart Law; Ampere's Law; Magnetic Field of a Solenoid; Magnetic Flux.	30	B1
Faraday's Law: Faraday's Law of Induction; Motional EMF; Lenz's Law.	31	B1
Temperature: Law of Thermodynamics and Temperature; Thermometers and Temperature Scales; The Constant-Volume Gas Thermometer and the Absolute Temperature Scale; Thermal Expansion of Solids and Liquids; The Ideal Gas Law.	19	B1
The 1st Law of Thermodynamics: Heat and Internal Energy; Heat Capacity; Latent Heat; Work and Heat; The 1st Law and Energy Conservation; Energy Transfer Mechanisms.	20	B1
The Kinetic Theory of Gases: Molecular Model of Ideal Gas; Specific Heat of Ideal Gas; Adiabatic Processes for Ideal Gas; The Equipartition of Energy.	21	B1
The 2nd Law of Thermodynamics: Heat Engines; Reversible and Irreversible Processes; Entropy and the 2nd Law; Entropy Changes in Irreversible Processes.	22	B1

MA 1230	**Enriched Calculus 2**	**3L:2T:0P**	**3 Credits**

Course Description: This course offers an in-depth study of single-variable integral calculus and its applications to provide a strong foundation for further study in mathematical disciplines. It is a required course for students in the engineering transfer program.

Prerequisites: A minimum of grade “C” or better in MATH 1130.

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

1. Compute integrals of a number of elementary functions by hand
2. Compute integrals to any desired accuracy by numerical methods;
3. Have an understanding of the fundamental nature of integration;
4. Be able to solve problems involving areas, volumes, lengths of curves, applied problems as centres of mass, work, fluid pressure and moments of inertia. etc., in which integration is used.

Texts/Materials

B1: W. Briggs, L. Cochran, B. Gillett, Calculus: Early Transcendentals, Single Variable, 2nd Edition, Pearson Education, 2015.

Course Topics	**Textbook Mapping**	
	Chapter	**Book**
Antiderivatives and Initial Value Problems, Velocity and Displacement, Sigma Notation and Reimann Sums, Evaluation of Integrals, Fundamental Theorem of Calculus and Area, Average value of a Function, Integration by Substitution, Area Between Curves, Volumes, Arc Length and Surface Area, force and Work, Moments, centroids, Centres of Mass, Logarithms and Exponential Models	5 & 6	B1
Basic Approaches, Integration by Parts, Trigonometric Integrals including Inverse Trig, Trig Substitution, Rational Functions and Partial Fractions, Other Integration Strategies, Numerical integration, Improper Integrals	7	B1
Basic Idea of Differential Equations, Direction Fields, Separable Differential Equations, First-Order Linear, Differential Equations, Modeling with Differential	8	B1

Equations		
Sequences and Series	9 & 10	B1
Polar Coordinates	11	B1

MA 1700	Discrete Mathematics	3L:1T:0P	3 Credits

Course Description: This course is an introduction to the foundation of modern mathematics including basic set theory; counting; solution to recurrence relations; logic and quantifiers; properties of integers; mathematical induction; asymptotic notation; introduction to graphs and trees; finite state machines and formal languages; Boolean algebra.

Prerequisites: A minimum grade of C+ in Principles of Math 12 or MATH 100 or a minimum grade of C+ in MATH 061 within the last two years at least or permission of instructor.

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

Introduce students to discrete mathematics including formal proof.

Express logical statements formally.

Construct a variety of mathematical proofs.

Evaluate functions and relations.

Use graphs to model and solve problems.

Analyze algorithms using appropriate notation to determine their efficiency.

Texts/Materials

B1: Epp, Susanna S., Discrete Mathematics with Applications, 4th ed, 2004.

Course Topics	**Textbook Mapping**	
	Chapter	**Book**
The Logic of Compound Statements	2	B1
The Logic of Quantified Statements	3	B1
Elementary Number Theory and Methods of Proof	4	B1
Mathematical Induction and Recursion	5	B1

Set Theory	6	B1
Functions	7	B1
Relations	8	B1
Counting and Probability	9	B1
Graphs and Trees	10	B1
Analysis of Algorithm Efficiency (if time permits)	11	B1
Regular Expressions and Finite-State Automata	12	B1

CH 1520	**Principles of Chemistry**	**3L:0T:3P**	**3T+1P Credits**

Course Description: The topics include gas laws, thermochemistry, equilibrium, entropy and free energy, redox reactions, and electrochemistry. Students are expected to become familiar with all these topics during the course. The laboratory stresses basic precision techniques in quantitative analytical chemistry as well as the use of analytical instrumentation and experiments in physical chemistry.

Prerequisites: Chemistry 12 or CHEM 0600 (a grade of B or better is recommended) and CHEM 1500 (C-minimum) or acceptance into the TRU Engineering program.

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

1. Understand, as well as be able to explain and apply the fundamental qualitative and quantitative properties of gases, including how gases are described by the Ideal gas equation, the origins of the ideal gas equation, the concepts of mole fraction, gaseous diffusion, non-ideal behavior of gasses and the assumptions used in kinetic molecular theory.
2. Understand, as well as be able to explain and apply the fundamental qualitative and quantitative aspects of thermodynamics including the first law of thermodynamics, various thermodynamic systems, calculations involving specific heat, heat transfer in calorimetry experiments, Hess's law and heats of solution.
3. Understand, as well as be able to explain and apply the fundamental qualitative and quantitative aspects of acid-base equilibria including dynamic equilibria, the equilibrium constant, various definitions of acids and bases, the meaning of Kw, calculations involving weak acids and bases, the qualitative and quantitative properties of strong acids, as well as those of polyprotic acids.
4. Understand, as well as be able to explain and apply the fundamental qualitative and quantitative aspects of buffers, titrations and solubility equilibria.
5. Understand, as well as be able to explain and apply the fundamental qualitative and quantitative aspects of entropy, free energy and electrochemistry as related to chemical reactions.
6. Understand, as well as be able to explain and apply the fundamental qualitative and quantitative aspects of kinetics as related to chemical reactions.

Texts/Materials

B1: R. Chang and K. Goldsby, Principles of Chemistry, McGraw-Hill, 2013.

Course Topics	**Textbook Mapping**	
	Chapter	**Book**
Section 1. Gases (6 lectures)	5	B1
Gas properties and pressure, The gas laws, The ideal gas law: Applications, Dalton's law of partial pressures, The kinetic molecular theory of gases, Diffusion and effusion, Real gases		
Section 2: Thermochemistry (5 lectures)	6	B1
Basic concepts, Enthalpy of chemical reactions, Calorimetry, Hess' law: Applications, Examples of enthalpy changes that refer to specific processes		
Section 3: Chemical equilibrium (2 lectures; Review)	14	B1
Principles of chemical equilibrium, Relationships involving equilibrium constants, Le Chatelier's Principle		
Section 4: Acid-Base equilibrium (5 lectures; Review)	15	B1
Acid-Base concepts, Acidity of a solution, Acid-base strength and equilibrium, Problems involving weak-acid and weak-base equilibria, Acid- base properties of salt solutions (Hydrolysis), Lewis theory of acids and bases		
Section 5: Buffers, Titrations & Solubility Equilibria (5 lectures)	16	B1
Buffer solutions: pH calculations, Acid-base titrations, Solubility equilibria		
Section 6: Entropy, Free Energy & Electrochemistry (8 lectures)	17, 18	B1
Spontaneous and non-spontaneous processes, Entropy and entropy changes, The Gibbs free energy, Thermodynamics of redox reactions		

DR 1520	Engineering Graphics	2L:0T:3P	3 Credits

Course Description: This course offers an introduction to learning and applying the principles of engineering graphical visualization and representation. The topics of this course include orthographic projection, technical sketching, development of the ability to visualize in three dimensions, engineering graphic standards and conventions, graphical presentation of engineering data, computer based graphics aids, engineering geometry and the solution of space problems.

Prerequisites: Admission to Engineering Program or written consent of Program Coordinator.

Educational Objectives/Outcomes:

After completing this course, the students will have the ability to:

1. Create orthographic, isometric and oblique sketches of mechanical parts;
2. Create fully dimensioned orthographic computer-aided two-dimensional drawings of mechanical parts;
3. Create sectional drawings of mechanical parts;
4. Create auxiliary views of mechanical parts;
5. Create three dimensional solid models of mechanical parts and extract two- dimensional orthographic views;
6. Solve engineering graphics problems using both descriptive geometry and the AutoCAD geometric calculator;
7. Form the concept of technical drawings for further engineering study and practice.

Texts/Materials

B1: Giesecke, et al., Technical Drawing with Engineering Graphics, 15th Edition (ISBN 0134306414)

Course Topics
Fundamental AutoCAD Commands, getting started with AutoCAD, system and dimensioning variables, lines, circles, arcs, polylines, and text entity creation, grid, snap and other drawing aids, object snap modes, editing commands, display commands, inquiry commands, blocks, plotting
Sketching and Text, alphabet of lines and line precedence sketching techniques, pictorial views, isometric sketches oblique sketches, perspective sketches, lettering
Orthographic Views, principle orthographic views, 3rd angle projection, missing views

- visualization
Sectional Views, extracting section views - types of sections hatch patterns, conventional breaks
Auxiliary Views, extracting auxiliary views, primary auxiliary views, secondary auxiliary views
Dimensioning and Tolerances, dimensioning practice, location dimensions, size dimensions, finish symbols, dimensioning system variables, dimension styles, tolerances
Threads and Fasteners, thread terminology, thread call-ups bolts and nuts, other types of fasteners
Fundamentals of 3D Drawing, paper space and model space UCS and WCS, display controls
Solid Modeling, wire frame, surface modeling, solid modeling solid entity creation, Boolean operations, regions, editing commands, display commands, mass properties
Descriptive Geometry, points in space, lines in space, planes in space, key views
Problem Solving, graphical problem solving techniques, civil problems, iterative solutions to engineering problems, solving and back solving formulas.

Third Semester

Course Code: SE301-19 **Course Title:** Data Structure & Algorithms **3L:1T:0P 3 Credits**

Detailed Contents:

Module 1: Introduction

Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off.
Searching: Linear Search and Binary Search Techniques and their complexity analysis.

[6 hrs] (CO1)

Module 2: Stacks and Queues

ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.

[10 hrs] (CO2, CO4, CO5)

Module 3: **Linked Lists**

Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: All operations their algorithms and the complexity analysis.
Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.

[10 hrs] (CO2, CO4, CO5)

Module 4: Sorting and Hashing

Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.

[10 hrs] (CO3)

Module 4: Graph

Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

[6 hrs] (CO2, CO4)

Course Outcomes:

The student will be able to:

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness;
2. Student will be able to handle operation like searching, insertion, deletion, traversing on various Data Structures and determine time and computational complexity;
3. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity;
4. Students will be able to choose appropriate Data Structure as applied to specific problem definition; &

5. Demonstrate the reusability of Data Structures for implementing complex iterative problems.

Suggested Books:

1. “Classic Data Structures”, Samanta and Debasis, 2nd edition, PHI publishers.
2. “Fundamentals of Data Structures”, Illustrated Edition by Ellis Horowitz, SartajSahni, Computer Science Press.
3. “Data Structures with C (Schaum's Outline Series)”, Seymour Lipschutz, 1st edition,McGraw Hill Education.

Reference Books:

1. Algorithms, Data Structures, and Problem Solving with C++”, Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company.
2. “How to Solve it by Computer”, 2nd Impression by R. G. Dromey, Pearson Education.

Course Code: SE302-19	**Course Title: Computer Networks**	**3L:1T:0P**	**3Credits**

Detailed Contents:

Module 1: Data Communication Components

Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum. **[8 hrs] (CO5)**

Module 2: Data Link Layer and Medium Access Sub Layer

Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, SlottedALOHA,CSMA/CDCDMA/CA. **[10 hrs] (CO5)**

Module 3: Network Layer

Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols. **[8 hrs] (CO5)**

Module 4: Transport Layer

Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm. **[8 hrs] (CO5)**

Module 5: Application Layer

Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography. **[8 hrs] (CO5)**

Course Outcomes:

The student will be able to:

CO1. Explain the functions of the different layer of the OSI Protocol;
CO2. Describe the function of each block of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs);
CO3. Develop the network programming for a given problem related TCP/IP protocol; &
CO4. Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

Suggested Books

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.

Reference Books

1. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
3. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

Course Code: SE304-19	**Course Title:** Data Structure & Algorithms Lab	**0L:0T:4P**	**2Credits**

List of Experiment:

Task 1: Write a program to insert a new element at end as well as at a given position in an array.

Task 2: Write a program to delete an element from a given whose value is given or whose position is given.

Task 3: Write a program to find the location of a given element using Linear Search.

Task 4: Write a program to find the location of a given element using Binary Search.

Task 5: Write a program to implement push and pop operations on a stack using linear array.

Task 6: Write a program to convert an infix expression to a postfix expression using stacks.

Task 7: Write a program to evaluate a postfix expression using stacks.

Task 8: Write a recursive function for Tower of Hanoi problem.

Task 9: Write a program to implement insertion and deletion operations in a queue using linear array.

Task 10: Write a menu driven program to perform following insertion operations in a single linked list:

i. Insertion at beginning
ii. Insertion at end
iii. Insertion after a given node
iv. Traversing a linked list

Task 11: Write a menu driven program to perform following deletion operations in a single linked list:

i. Deletion at beginning
ii. Deletion at end
iii. Deletion after a given node

Task 12: Write a program to implement push and pop operations on a stack using linked list.

Task 13: Write a program to implement push and pop operations on a queue using linked list.

Task 14: Program to sort an array of integers in ascending order using bubble sort.

Task 15: Program to sort an array of integers in ascending order using selection sort.

Task 16: Program to sort an array of integers in ascending order using insertion sort.

Task 17: Program to sort an array of integers in ascending order using quick sort.

Task 18: Program to traverse a Binary search tree in Pre-order, In-order and Post-order.

Task 19: Program to traverse graphs using BFS.

Task 20: Program to traverse graphs using DFS.

Lab Outcomes:

The student will be able to:

1. Improve practical skills in designing and implementing basic linear data structure algorithms;
2. Improve practical skills in designing and implementing Non-linear data structure algorithms;
3. Use Linear and Non-Linear data structures to solve relevant problems;
4. Choose appropriate Data Structure as applied to specific problem definition; &
5. Implement Various searching algorithms and become familiar with their design methods.

Reference Books:

1. “Data Structures with C (Schaum's Outline Series)”, Seymour Lipschutz, 1st edition, McGraw Hill Education.

Course Code: SE305-19	**Course Title: Computer networks** Lab	**0L:0T:2P**	**1Credits**

List of Experiments:

Task1 : To study the different types of Network cables and network topologies

Task2 :Practically implement and test the cross-wired cable and straight through cable using clamping tool and network lab cable tester.

Task3 :Study and familiarization with various network devices.

Task4 :Familiarization with Packet Tracer Simulation tool/any other related tool. Task5 :Study and Implementation of IP Addressing Schemes

Task6 :Creation of Simple Networking topologies using hubs and switches

Task7 :Simulation of web traffic in Packet Tracer

Task8 :Study and implementation of various router configuration commands

Task9 :Creation of Networks using routers.

Task10 :Configuring networks using the concept of subnetting

Task11 :Practical implementation of basic network command and Network configuration commands like ping, ipconfig, netstat , tracert etc. for trouble shooting network related problems.

Task12 :Configuration of networks using static and default routes.

Course Outcomes:

The students will be able to

1: Know about the various networking devices, tools and also understand the implementation of network topologies;

2:Create various networking cables and know how to test these cables;

3:Create and configure networks in packet tracer tool using various network devices and topologies;

4:Understand IP addressing and configure networks using the subnettin;

5:Configure routers using various router configuration commands;&

6:Troubleshoot the networks by using various networking commands.

Course Code:BTES301-18	**Course Title: Digital Electronics**	**3L:0T:0P**	**3Credits**

Detailed Contents: Module 1:

NUMBER SYSTEMS: Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1's, 2's complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII.
LOGIC GATES: AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations of Logic Functions using gates, NAND-NOR implementations.

Module 2 :
BOOLEAN ALGEBRA: Boolean postulates and laws – De-Morgan's Theorem, Principle of Duality, Boolean expression – Boolean function, Minimization of Boolean expressions – Sum of Products (SOP), Product of Sums (POS), Minterm, Maxterm, Canonical forms, Conversion between canonical forms, Karnaugh map Minimization, Don't care conditions, Quine-McCluskey method.

Module 3:
COMBINATIONAL CIRCUITS: Design procedure – Adders, Subtractors, BCD adder, Magnitude Comparator, Multiplexer/Demultiplexer, encoder/decoder, parity checker, code converters. Implementation of combinational logic using MUX, BCD to
7 segment decoder.

SEQUENTIAL CIRCUITS: Flip flops SR, JK, T, D and Master slave, Excitation table, Edge triggering, Level Triggering, Realization of one flip flop using other flip flops. Asynchronous/Ripple counters, Synchronous counters, Modulo-n counter, Ring Counters. Design of Synchronous counters: state diagram, Circuit implementation. Shift registers.

Module 4:
MEMORY DEVICES: Classification of memories, RAM organization, Write operation, Read operation, Memory cycle. ROM organization, PROM, EPROM, EEPROM, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA).

A/D & D/A CONVERTORS : Analog & Digital signals. sample and hold circuit, A/D and D/A conversion techniques (Weighted type, R-2R Ladder type, Counter Type, Dual Slope type, Successive Approximation type).

COURSE OUTCOME:At the end of course the student will be able to:

1. Demonstrate the operation of simple digital gates, identify the symbols, develop the truth table for those gates; combine simple gates into more complex circuits; change binary, hexadecimal, octal numbers to their decimal equivalent an vice versa.
2. Demonstrate the operation of a flip-flop. Design counters and clear the concept of shift registers.
3. Study different types of memories and their applications.Convert digital signal into analog and vice versa.

Suggested Readings/ Books:

□ 1. Morris Mano, **Digital Design**, Prentice Hall of India Pvt. Ltd

□ 2. Donald P.Leach and Albert Paul Malvino, **Digital Principles and Applications**, 5 ed., Tata McGraw HillPublishing Company Limited, New Delhi, 2003.

□ 3. R.P.Jain, **Modern Digital Electronics**, 3 ed., Tata McGraw–Hill publishing company limited, New Delhi, 2003.

- 4. Thomas L. Floyd, **Digital Fundamentals**, Pearson Education, Inc, New Delhi, 2003
- 5.Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, **Digital System - Principles and Applications**, Pearson Education.
- 6. Ghosal ,**Digital Electronics**, Cengage Learning.

Course Code:BTES302-18	**Course Title: Digital Electronics Lab**	**0L:0T:2P**	**1Credits**

List of Experiments:

1. To verify the Truth-tables of all logic gates.
2. To realize and verify the Half & full adder circuits using logic gates.
3. To realize Half & full subtractor circuits using logic gates.
4. To realize Encoder and Decoder circuits
5. To realize Multiplexer circuits
6. To realize 4-bit binary-gray & gray-binary converters.
7. To realize comparator circuit for two binary numbers of 2-bit each.
8. To realize Full adder & full subtractor circuits using encoder.
9. To design Full adder & full subtractor circuits using multiplexer.
10. To design and verify the Truth tables of all flip-flops.
11. To design Mod-6/Mod-9 synchronous up-down counter.

Course Outcomes

At the end of this course student will demonstrate the ability to:

1. Realize combinational circuits using logic gates.
2. Realize sequential circuits using logic gates.
3. Realize various types of Flip-flops and counters

Course Code: HSMC101-18	**Course Title: Development of Societies**	**3L:0T:0P**	**3Credits**

<u>Detailed Contents:</u>

Unit I: Social Development (5 hours)

1. Concepts behind the origin of Family, Clan and Society
2. Different Social Systems
3. Relation between Human being and Society
4. Comparative studies on different models of Social Structures and their evolution

Unit II: Political Development (3 hours)

1. Ideas of Political Systems as learnt from History
2. Different models of Governing system and their comparative study

Unit III: Economic Development (18 hours)

1. Birth of Capitalism, Socialism, Marxism
2. Concept of development in pre-British, British and post British period- Barter, Jajmani
3. Idea of development in current context.
4. E. F. Schumacher's idea of development, Buddhist economics. Gandhian idea of development. Swaraj and Decentralization. **PROJECT: Possible projects in this course could be**

a) Interact with local communities and understand their issues.

b) Study local cottage industry and agricultural practices. Role of engineering and specialized knowledge.

c) Evaluation of technology in the context of its application. Social impact of technology. Environmental impact of technology. Evaluation from a holistic perspective.

Course Code: HSMC102-18	**Course Title: PHILOSOPHY**	**3L:0T:0P**	**3Credits**

Detailed Contents: **Unit 1:**

The difference between knowledge (Vidya) and Ignorance (Avidya):

a. Upanishads;

b. Six systems orthodox and Heterodox Schools of Indian Philosophy. c. Greek Philosophy:

Unit 2:

Origin of the Universe:

- NasidiyaSukta: "Who really knows?”
- Brhadaranyaka Upanishad; Chandogya Upanishad: Non-self, Self, real and unreal.
- Taittiriya Upanishad: SikshaValli.
- Plato’s Symposium: Lack as the source of desire and knowledge.
- Socratic’s method of knowledge as discovery.
- Language: Word as root of knowledge (Bhartrahari’sVakyapadiyam)
- Fourteen Knowledge basis as a sources of Vidya: Four Vedas; Six auxiliary sciences (Vedangas); Purana, Nyaya, Mimamsa and Dharma Sastras.

Unit 3:

Knowledge as Power: Francis Bacon. Knowledge as both power and self-realization in Bagavad Gita.

Unit 4:

Knowledge as oppression: M. Foucault. Discrimination between Rtam and Satyam in Indian Philosophy.

Unit 5:

Knowledge as invention: Modern definition of creativity; scientific activity in the claim that science invents new things at least through technology.

Unit 6:

Knowledge about the self, transcendental self; knowledge about society, polity and nature.

Unit 7:

Knowledge about moral and ethics codes.

Unit 8:

Tools of acquiring knowledge: Tantrayuktis, a system of inquiry (Caraka, Sushruta, Kautilya, Vyasa)

READINGS

1. Copleston, Frederick, History of Philosophy, Vol. 1.Great Britain: Continuum.

2 Hiriyanna, M. Outlines of Indian Philosophy, MotilalBanarsidass Publishers; Fifth Reprint edition (2009)

3 Sathaye, Avinash, Translation of NasadiyaSukta

4. Ralph T. H. Griffith. The Hymns of the Ṛgveda. MotilalBanarsidass: Delhi: 1973.

5. Raju, P. T. Structural Depths of Indian Thought, Albany: State University of New York

Press.
6. Plato, Symposium, Hamilton Press.
7. KautilyaArtha Sastra. Penguin Books, New Delhi.
8. Bacon, Nova Orgum
9. Arnold, Edwin. The Song Celestial.
10. Foucault, Knowledge/Power.
11. Wildon, Anthony, System of Structure.
12. Lele, W.K. The Doctrine of Tantrayukti. Varanasi: Chowkamba Series.
13. Dasgupta, S. N. History of Indian Philosophy, MotilalBanasidas, Delhi.
14. Passmore, John, Hundred Years of Philosophy, Penguin.

ASSESSMENT (indicative only):

Ask students to do term papers, for example, writing biographical details of founders, sustainers, transmitters, modifiers, rewriters; translating monographs of less known philosophers such as K. C. Bhattacharys, Daya Krishna, Gopinath Bhattacharya; comparative study of philosophical system such as MadhyasthaDarshan.

OUTCOME OF THE COURSE:

Students will develop strong natural familiarity with humanities along with right understanding enabling them to eliminate conflict and strife in the individual and society. Students shall be able to relate philosophy to literature, culture, society and lived experience can be considered.

Course Code: SE303-18	**Course Title: Software Engineering**	**3L:0T:0P**	**3Credits**

Detailed Contents:

UNIT1:

Evolution and impact of Software engineering, software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification. **8 hrs.**

UNIT2*:* Basic issues in software design, modularity, cohesion, coupling and layering, function-oriented software design: DFD and Structure chart, object modeling using UML, Object-oriented software development, user interface design. Coding standards and Code review techniques. **6hrs**

UNIT3*:* Fundamentals of testing, White-box, and black-box testing, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics, reliability growth modeling. **8 hrs.**

UNIT4*:* Software project management, Project planning and control, cost estimation, project scheduling using PERT and GANTT charts, cost-time relations: Rayleigh-Norden results, quality management **8 hrs.**

UNIT 5: ISO and SEI CMMI, PSP and Six Sigma. Computer aided software engineering, software maintenance, software reuse, Component-based software development. **6 hrs**

Suggested Readings/ Books:

1. Roger Pressman, **"Software Engineering: A Practitioners Approach,**(6th Edition), McGraw Hill, 1997.

2. Sommerville,"**Software Engineering, 7th edition**", Adison Wesley, 1996.

3. Watts Humphrey,**" Managing software process**", Pearson education, 2003.

4. James F. Peters and Witold Pedrycz, **" Software Engineering – An Engineering Approach"**, Wiley.

5. Mouratidis and Giorgini. **"Integrating Security and Software Engineering–Advances and Future"**,

IGP. ISBN – 1-59904-148-0.

6. Pankaj Jalote, **"An integrated approach to Software Engineering"**, Springer/Narosa.

Course Code: SE306-18	**Course Title: Software Engineering lab**	**0:0T:P**	**3Credits**

Detailed List of Tasks:

1. Study and usage of OpenProj or similar software to draft a project plan

2. Study and usage of OpenProj or similar software to track the progress of a project

3. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents for some problems

4. Preparation of Software Configuration Management and Risk Management related documents

5. Study and usage of any Design phase CASE tool

6. To perform unit testing and integration testing

7. To perform various white box and black box testing techniques

8. Testing of a web site

Suggested Tools - Visual Paradigm, Rational Software Architect. Visio, Argo UML, Rational Application Developer etc. platforms.

Fourth Semester

Course Code: SE401-19	Course Title: Database management Systems	L:3; T:0; P:0	3Credits

Detailed contents

Module1:Database system architecture
Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).
Data models: Entity-relationship model, network model, relational and object oriented Data models, integrity constraints, data manipulation operations.

Module2: Relational query languages
Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE,DB2,SQL server.
Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation,Lossless design.
Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

Module3:Storage strategies
Indices, B-trees, hashing.

Module4:
Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Controlschemes, Databaserecovery.

Module5:
Database Security: Authentication, Authorization and accesscontrol, DAC, MAC and RBAC models, Intrusion detection, SQLinjection.

Module6:Advanced topics
Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

Suggested books:

1. “Database System Concepts”, 6th Edition by AbrahamSilberschatz,HenryF. Korth,S.Sudarshan, McGraw-Hill.

Suggested reference books

1 “PrinciplesofDatabaseandKnowledge–BaseSystems”,Vol1byJ.D.Ullman, Computer Science Press.
2 “FundamentalsofDatabaseSystems”,5thEditionbyR.ElmasriandS.Navathe, PearsonEducation
3 “FoundationsofDatabases”, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu,Addison-Wesley

CourseOutcomes

CO1. For a given query write relational algebra expressions for that query and optimize the Developed expressions
CO2. For a given specification of the requirement design the databases using ER method and normalization.

CO3. ForagivenspecificationconstructtheSQLqueriesforOpensourceandCommercial DBMS-MYSQL, ORACLE,andDB2.
CO4. ForagivenqueryoptimizeitsexecutionusingQueryoptimizationalgorithms
CO5. For a given transaction-processing system, determine the transaction atomicity, consistency,isolation, and durability.
CO6. Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

Course Code: SE402-19	Course Title: Computer Organisation and Architecture	3L:1T:0P	3Credits

Pre-requisites: Digital Electronics

Detailed Contents:

Module 1: 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 set of 8085 processor.

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-andadd, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

[10 hrs] (CO1, CO2)

Module 2: Introduction to x86 architecture.

CPU control unit design: Hardwired and micro-programmed design approaches, Case study – design of a simple hypothetical CPU.

Memory system design: semiconductor memory technologies, memory organization. **Peripheral devices and their characteristics**: Input-output subsystems, I/O device interface, 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, I/O device interfaces – SCII, USB.

[12 hrs] (CO2, CO4)

Module 3: Pipelining

Basic concepts of pipelining, throughput and speedup, pipeline hazards.

Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

[10 hrs] (CO5)

Module 4: Memory Organization

Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

[10 hrs] (CO3)

Course Outcomes:

The student will be able to:

1. Understand functional block diagram of microprocessor;
2. Apply instruction set for Writingassembly language programs;
3. Design a memory module and analyze its operation by interfacing with the CPU;
4. Classify hardwired and microprogrammed control units; &

5. Understand the concept of pipelining and its performance metrics.

Suggested Books:

1. “ComputerOrganization and Architecture”, Moris Mano,
2. “ComputerOrganization and Design: The Hardware/Software Interface”, 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
3. “Computer Organization and Embedded Systems”, 6th Edition by CarlHamacher, McGraw Hill Higher Education.

Reference Books:

1. “Computer Architecture and Organization”, 3rd Edition by John P. Hayes, WCB/McGraw-Hill
2. “Computer Organization and Architecture: Designing for Performance”, 10th Edition by William Stallings, Pearson Education.
3. “Computer System Design and Architecture”, 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Course Code: SE403-19	**Course Title:** Operating Systems	**3L:1T:0P**	**3Credits**

Detailed Contents:

Module1: Introduction

Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

[6 hrs] (CO1)

Module2: Processes

Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching

Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

[10 hrs] (CO2, CO3)

Module 3: Inter-process Communication

Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson’s Solution, TheProducer\ConsumerProblem, Semaphores, EventCounters, Monitors, Message Passing, Classical IPC Problems: Reader’s & Writer Problem, Dinning Philosopher Problem etc.

[8 hrs] (CO2)

Module 4: Deadlocks

Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker’s algorithm, Deadlock detection and Recovery.

[8 hrs] (CO3)

Module5: MemoryManagement

Basicconcept,LogicalandPhysical address map, Memory allocation: Contiguous Memory allocation –Fixedandvariable partition–Internaland External fragmentation and Compaction; Paging:

Principle of operation – Page allocation–Hardware support for paging, Protection and sharing, Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

[10 hrs] (CO4)

Module 6: I/O Hardware

I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocationmethods (contiguous, linked, indexed), Free Space Management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

[8 hrs] (CO5, CO6)

<u>Course Outcomes:</u>

The student will be able to:

1. Explain basic operating system concepts such as overall architecture, system calls, user mode and kernel mode;
2. Distinguish concepts related to processes, threads, process scheduling, race conditions and critical sections;
3. Analyze and apply CPU scheduling algorithms, deadlock detection and prevention algorithms;
4. Examine and categorize various memory management techniques like caching, paging, segmentation, virtual memory, and thrashing;
5. Design and implement file management system; &
6. Appraise high-level operating systems concepts such as file systems, disk-scheduling algorithms and various file systems.

<u>Suggested Books:</u>

1. Operating System Concepts Essentials, 9th Edition by AviSilberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

<u>Reference Books:</u>

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

----

Course Code: BTSE404-19	**Course Title:** Design and Analysis of Algorithms	**3L:0T:0P**	**3Credits**

Pre-requisites: Data Structures

Detailed Contents:

Module1: Introduction

Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

[8 hrs] (CO1)

Module 2: Fundamental Algorithmic Strategies

Brute-Force, Greedy,Dynamic Programming, Branch- and-Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving: Bin Packing, Knap Sack, TSP.**[10 hrs] (CO1, CO2)**

Module 3: Graph and Tree Algorithms

Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm. **[10 hrs] (CO3)**

Module 4: Tractable and Intractable Problems

Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard.

Cook's theorem, Standard NP-complete problems and Reduction techniques.**[8 hrs] (CO5)**

Module 5: Advanced Topics

Approximation algorithms, Randomized algorithms, Heuristics and their characteristics.

[6 hrs] (CO1, CO4, CO5)

Course Outcomes:

The student will be able to:

1. For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms;
2. Explain when an algorithmic design situation calls for which design paradigm (greedy/ divide and conquer/backtrack etc.);
3. Explain model for a given engineering problem, using tree or graph, and writethe corresponding algorithm to solve the problems;
4. Demonstrate the ways to analyze approximation/randomized algorithms (expected running time, probability of error); &
5. Examine the necessity for NP class based problems and explain the use of heuristic techniques.

Suggested Books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Data Structures and Algorithms in C++, Weiss, 4th edition, Pearson.
3. Fundamentals of Computer Algorithms – E. Horowitz, Sartaj Saini, Galgota Publications.

Reference Books

1. Algorithm Design, 1stEdition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
3. Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Course Code: SE406-19	**Course Title:** Computer Organization & ArchitectureLab	**0L:0T:2P**	**1Credits**

List of Experiment:

Task 1: Computer Anatomy- Memory, Ports, Motherboard and add-on cards.
Task 2: Dismantling and assembling PC.
Task 3: Introduction to 8085 kit.
Task 4: 2. Addition of two 8 bit numbers, sum 8 bit.
Task 5: Subtraction of two 8 bit numbers.
Task 6: Find 1’s complement of 8-bit number.
Task 7: Find 2’s complement of 8-bit number.
Task 8: Shift an 8-bit no. by one bit.
Task 9: Find Largest of two 8 bit numbers.
Task 10: Find Largest among an array of ten numbers (8 bit).
Task 11: Sum of series of 8 bit numbers.
Task 12: Introduction to 8086 kit.
Task 13: Addition and subtraction of two 16 bit numbers, sum 16 bit.
Task 14: Implement of Booth’s algorithm for arithmetic operations.
Task 15: Find 1’s and 2’s complement of 16-bit number.
Task 16: Implement simple programs using I/O based interface.

Lab Outcomes:

The student will be able to:

1. Assemble personal computer;
2. Implement the various assembly language programs for basic arithmetic and logical operations; &
3. Demonstrate the functioning of microprocessor/microcontroller based systems with I/O interface.

Reference Books:

1. Fundamentals of Microprocessors and Microcontrollersby B. Ram, Dhanpat Rai

Course Code: SE407-19	**Course Title:** Operating Systems Lab	**0L:0T:4P**	**2Credits**

List of Experiment:

Task 1: Installation Process of various operating systems.

Task 2: Implementation of CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority.

Task 3: Virtualization, Installation of Virtual Machine Software and installation of Operating System on Virtual Machine.

Task 4: Commands for files & directories: cd, ls, cp, md, rm, mkdir, rmdir. Creating and viewing files using cat. File comparisons. Disk related commands: checking disk free spaces. Processes in linux, connecting processes with pipes, background processing, managing multiple processes. Background process: changing process priority, scheduling of processes at command, batch commands, kill, ps, who, sleep. Printing commands, grep, fgrep, find, sort, cal, banner, touch, file. File related commands ws, sat, cut, grep.

Task 5: Shell Programming: Basic of shell programming, various types of shell, Shell Programming in bash, conditional & looping statement, case statements, parameter passing and arguments, shell variables, shell keywords, creating shell programs for automate system tasks, report printing.

Task 6: Implementation of Bankers algorithm for the purpose of deadlock avoidance.

Lab Outcomes:

The student will be able to:

1. Understand and implement basic services and functionalities of the operating system;
2. Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority;
3. Implement commands for files and directories;
4. Understand and implement the concepts of shell programming;
5. Simulate file allocation and organization techniques; &
6. Understand the concepts of deadlock in operating systems and implement them in multiprogramming system.

Reference Books:

1. Operating Systems: Design and Implementation, Albert S. Woodhull and Andrew S. Tanenbaum, Pearson Education.

Course Code: SE408-19	**Course Title:** Design and Analysis of Algorithms Lab	**0L:0T:4P**	**2Credit**

List of Experiment:

Task 1: Code and analyze solutions to following problem with given strategies:

i. Knap Sack using greedy approach

ii. Knap Sack using dynamic approach

Task 2: Code and analyze to find an optimal solution to matrix chain multiplication using dynamic programming.

Task 3: Code and analyze to find an optimal solution to TSP using dynamic programming.

Task 4: Implementing an application of DFS such as:

i. to find the topological sort of a directed acyclic graph

ii. to find a path from source to goal in a maze.

Task 5: Implement an application of BFS such as:

i. to find connected components of an undirected graph

ii. to check whether a given graph is bipartite.

Task 6: Code and analyze to find shortest paths in a graph with positive edge weights using Dijkstra’s algorithm.

Task 7: Code and analyze to find shortest paths in a graph with arbitrary edge weights using Bellman-Ford algorithm.

Task 8: Code and analyze to find shortest paths in a graph with arbitrary edge weights using Flyods’ algorithm.

Task 9: Code and analyze to find the minimum spanning tree in a weighted, undirected graph using Prims’ algorithm

Task 10: Code and analyze to find the minimum spanning tree in a weighted, undirected graph using Kruskals’ algorithm.

Task 11: Coding any real world problem or TSP algorithm using any heuristic technique.

Lab Outcomes:

The student will be able to:

1. Improve practical skills in designing and implementing complex problems with different techniques;
2. Understand comparative performance of strategies and hence choose appropriate, to apply to specific problem definition;
3. Implement Various tree and graph based algorithms and become familiar with their design methods; &
4. Design and Implement heuristics for real world problems.

Reference Books

1. Data Structures and Algorithms in C++, Weiss, 4th edition, Pearson

2. Data Structures and Algorithms using Python and C++, David M. Reed and John Zelle, 2009 edition (available as e book), Franklin Beedle& Associates.

UNIVERSAL HUMAN VALUES 2: UNDERSTANDING HARMONY

Course code: HSMC122-18 Credits: 3

COURSE TOPICS:

The course has 28 lectures and 14 practice sessions in 5 modules:

Module 1: Course Introduction - Need, Basic Guidelines, Content and Process for Value Education

1. Purpose and motivation for the course, recapitulation from Universal Human Values-I
2. Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and ExperientialValidation- as the process for self-exploration.
3. Continuous Happiness and Prosperity- A look at basic Human Aspirations
4. Right understanding, Relationship and Physical Facility- the basic requirements for fulfilment of aspirations of every human being with their correct priority
5. Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario.
6. Method to fulfil the above human aspirations: understanding and living in harmony at various levels.

Include practice sessions to discuss natural acceptance in human being as the innate acceptance for living with responsibility (living in relationship, harmony and co-existence) rather than as arbitrariness in choice based on liking-disliking.

Module 2: Understanding Harmony in the Human Being - Harmony in Myself!

7. Understanding human being as a co-existence of the sentient 'I' and the material 'Body'
8. Understanding the needs of Self ('I') and 'Body' - happiness and physical facility
9. Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer)
10. Understanding the characteristics and activities of 'I' and harmony in 'I'
11. Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail
12. Programs to ensureSanyam and Health.

Include practice sessions to discuss the role others have played in making material goods available to me. Identifying from one's own life. Differentiate between prosperity and accumulation. Discuss program for ensuring health vs dealing with disease.

Module 3: Understanding Harmony in the Family and Society- Harmony in Human-Human Relationship

13. Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship.

14. Understanding the meaning of Trust; Difference between intention and competence
15. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship.
16. Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals.
17. Visualizing a universal harmonious order in society- Undivided Society, Universal Order- from family to world family.
Include practice sessions to reflect on relationships in family, hostel and institute as extended family, real life examples, teacher-student relationship, goal of education etc. Gratitude as a universal value in relationships. Discuss with scenarios. Elicit examples from students' lives.

Module 4: Understanding Harmony in the Nature and Existence - Whole existence as Coexistence
18. Understanding the harmony in the Nature
19. Interconnectedness and mutual fulfilment among the four orders of nature - recyclability and self-regulation in nature
20. Understanding Existence as Co-existence of mutually interacting units in all- pervasive space
21. Holistic perception of harmony at all levels of existence.
Include practice sessions to discuss human being as cause of imbalance in nature (film "Home" can be used), pollution, depletion of resources and role of technology etc.

Module 5: Implications of the above Holistic Understanding of Harmony on Professional Ethics
22. Natural acceptance of human values
23. Definitiveness of Ethical Human Conduct
24. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order
25. Competence in professional ethics: a. Ability to utilize the professional competence for augmenting universal human order b. Ability to identify the scope and characteristics of peoplefriendly and eco -friendly production systems, c. Ability to identify and develop appropriatetechnologies and management patterns for above production systems.
26. Case studies of typical holistic technologies, management models and production systems.
27. Strategy for transition from the present state to Universal Human Order: a. At the level of individual: as socially and ecologically responsible engineers, technologists and managers b. At the level of society: as mutually enriching institutions and organizations.
28. Sum up. Include practice Exercises and Case Studies will be taken up in Practice (tutorial) Sessions eg. to discuss the conduct as an engineer or scientist etc.

3. READINGS:

3.1 Text Book

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 2010.

3.2 Reference Books

1. Jeevan Vidya: EkParichaya, A. Nagaraj, Jeevan VidyaPrakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J CKumarappa
8. Bharat Mein Angreji Raj -PanditSunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)

OUTCOME OF THE COURSE:

By the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature); they would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind. They would have better critical ability. They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society). It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

This is only an introductory foundational input. It would be desirable to follow it up by

a) Faculty -student or mentor-mentee programs throughout their time with the institution.
b) Higher level courses on human values in every aspect of living. E.g. as a professional.

Course Code: SE405-19	**Course Title: Database management System lab**	**0:0T:4P**	**4Credits**

Details of Tasks:

1. Introduction to SQL and installation of SQL Server / Oracle.

2. Data Types, Creating Tables, Retrieval of Rows using Select Statement, Conditional Retrieval of Rows, Alter and Drop Statements.

3. Working with Null Values, Matching a Pattern from a Table, Ordering the Result of a Query, Aggregate Functions, Grouping the Result of a Query, Update and Delete Statements.

4. Set Operators, Nested Queries, Joins, Sequences.

5. Views, Indexes, Database Security and Privileges: Grant and Revoke Commands, Commit and Rollback Commands.

6. PL/SQL Architecture, Assignments and Expressions, Writing PL/SQL Code, Referencing Non-SQL parameters.

7. Stored Procedures and Exception Handling.

8. Triggers and Cursor Management in PL/SQL.

Suggested Tools – MySQL, DB2, Oracle, SQL Server 2012, Postgre SQL, SQL lite

Course Outcomes:

CO1: This practical will enable students to retrieve data from relational databases using SQL.

CO2: students will be able to implement generation of tables using datatypes

CO3: Students will be able to design and execute the various data manipulation queries.

CO4: Students will also learn to execute triggers, cursors, stored procedures etc.

Fifth Semester

BTES501-18	**Enterprise Resource Planning**	**3L:0T:0P**	**3 Credits**

Course Details:

UNIT 1 INTRODUCTION

ERP: An Overview, Enterprise – An Overview, Benefits of ERP, ERP and Related Technologies, Business Process Reengineering (BPR), Data Warehousing, Data Mining, OLAP, SCM [9hrs., CO1]

UNIT II ERP IMPLEMENTATION

ERP Implementation Lifecycle, Implementation Methodology, Hidden Costs, Organizing the Implementation, Vendors, Consultants and Users, Contracts with Vendors, Consultants and Employees, Project Management and Monitoring [9hrs., CO2]

UNIT III THE BUSINESS MODULES

Business modules in an ERP Package, Finance, Manufacturing, Human Resources, Plant Maintenance, Materials Management, Quality Management, Sales and Distribution [9hrs., CO3]

UNIT IV THE ERP MARKET

ERP Market Place, SAP AG, Peoplesoft, Baan, JD Edwards, Oracle, QAD, SSA [9hrs., CO4]

UNIT V ERP – PRESENT AND FUTURE

Turbo Charge the ERP System, EIA, ERP and e-Commerce, ERP and Internet, Future Directions

[6hrs., CO1]

TEXT BOOK

1. Alexis Leon, “ERP Demystified”, Tata McGraw Hill, New Delhi, 2000

REFERENCES

1. Joseph A Brady, Ellen F Monk, Bret Wagner, “Concepts in Enterprise Resource Planning”, Thompson Course Technology, USA, 2001.

2. Vinod Kumar Garg and Venkitakrishnan N K, “Enterprise Resource Planning – Concepts and Practice”, PHI, New Delhi, 2003

Course outcomes: The students at the end will be able;

CO1: To know the basics of ERP

CO2: To understand the key implementation issues of ERP

CO3: To know the business modules of ERP

CO4: To be aware of some popular products in the area of ERP

Course Code: SE501-19	Course Title : Agile Software Development	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: **Introduction**

Need of Agile software development, History of Agile, Agile context– manifesto, principles, methods, values. The benefits of agile in software development.

[**6hrs] (CO 1)**

UNIT 2: Agile Design Methodologies

Fundamentals, Design principles–Single responsibility, Open-closed, Liskov-substitution, Dependency-inversion, Interface-segregation. [**6hrs] (CO 2)**

UNIT 3: Scrum

Introduction to scrum framework, Roles: Product owner, team members and scrum master, Events: Sprint, sprint planning, daily scrum, sprint review, and sprint retrospective, Artifacts: Product backlog, sprint backlog and increments. User stories- characteristics and contents.

[**8hrs] (CO 3)**

UNIT 4: Kanban

Introduction to Kanban framework, Workflow, Limit the amount of work in progress, pulling work from column to column, Kanban board, Adding policies to the board, Cards and their optimization.

[6hrs] (**CO 4)**

UNIT 5: Extreme Programming

Basic values and principles, Roles, Twelve practices of XP, Pair programming, XP team, Life cycle and tools for XP. **[6hrs] (CO 5)**

UNIT 6: Agile Testing

The Agile lifecycle and its impact on testing, Test driven development– Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation. **[6hrs] (CO 6)**

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Understand concept of agile software engineering and its advantages in software development.

CO2 Explain the role of design principles in agile software design.

CO3 Define the core practices behind Scrum framework.

CO4 Understand key principles of agile software development methodology-Kanban.

CO5 Describe implications of functional testing, unit testing, and continuous integration.

CO6 Understand the various tools available to agile teams to test the project.

Suggested Readings/ Books:

1. Ken Schawber, Mike Beedle, “Agile Software Development with Scrum”, Pearson.
2. Robert C. Martin, “Agile Software Development, Principles, Patterns and Practices”, Prentice Hall.
3. Mike Cohn, “User Stories Applied: For Agile Software Development”, Addison Wesley Signature Series.
4. Lisa Crispin, Janet Gregory, “Agile Testing: A Practical Guide for Testers and Agile Teams”, Addison Wesley.
5. Paul VII, “Agile: The Complete Overview of Agile Principles and Practices (Agile Product Management)”.
6. Robert Martin, “Agile Software Development, Principles, Patterns, and Practices”, Pearson New International Edition.
7. Greene Jennifer,” Learning Agile”, O’Reilly Series.

Course Code: SE503-19	**Course Title : Agile Software Development Lab**	**L:0T:2P**	**Credits:1**

Detailed List of Tasks:

1. Understand the background and driving forces for taking an Agile Approach to Software Development.
2. Build out a backlog and user stories.
3. To study and use automated build tool.
4. To study-- version control tool.
5. To study Continuous Integration tool.
6. Apply Design principle and Refactoring to achieve agility.
7. Perform Testing activities within an agile project.

8. Mini Project: based on tools

Course Code: BTCS502-18	**Course Title:** Formal Language & Automata Theory	**3L:1T:0P**	**3Credits**	**42 Hours**

Detailed Contents

Module 1: Introduction

Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. **[3hrs] (CO1)**

Module 2: Regular languages and finite automata:

Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. **[8hrs] (CO2)**

Module 3: Context-free languages and pushdown automata

Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. **[8hrs] (CO3)**

Module 4: Context-sensitive languages

Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. **[5hrs] (CO4)**

Module 5: Turing machines

The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. **[8hrs] (CO 5)**

Module 6: Undecidability & Intractablity:

Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

Intractablity: Notion of tractability/feasibility. The classes NP and co-NP, their importance. Polynomial time many-one reduction. Completeness under this reduction. Cook-Levin theorem: NP-completeness of propositional satisfiability, other variants of satisfiability. NP-complete problems from other domains: graphs (clique, vertex cover, independent sets, Hamiltonian cycle), number problem (partition), set cover **[12hrs] (CO5)**

Course Outcomes: The student will be able to:

CO1: Write a formal notation for strings, languages and machines.

CO2: Design finite automata to accept a set of strings of a language.

CO3: Design context free grammars to generate strings of context free language .

CO4: Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars

CO5: Distinguish between computability and non-computability and Decidability and undecidability.

Text Books:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.

Reference Books:

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
2. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
3. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
4. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

Course Code: SE502-19	**Course Title:** Software Testing and Quality Assurance	**3L:0T:0P**	**3Credits**	**42 Hours**

Detailed Contents:

Module 1:

Testing as an engineering activity, Role of process in software quality, Testing as a process, Basic definitions, Software testing principles, The tester's role in a software development organization, Origins of defects, Defect classes, The defect repository and test design, Defect examples, Developer / Tester support for developing a defect repository.

[7hrs] (CO 1)

Module 2:

Testing techniques and levels of testing:

Using White Box Approach to Test design - Static Testing Vs. Structural Testing, Code Functional Testing, Coverage and Control Flow Graphs, Using Black Box Approaches to Test Case Design, Random Testing, Requirements based testing, Decision tables, State-based testing, Cause-effect graphing, Error guessing, Compatibility testing, Levels of Testing -Unit Testing, Integration Testing, Defect Bash Elimination. System Testing - Usability and Accessibility Testing, Configuration Testing, Compatibility Testing **[10hrs] (CO2)**

Module 3:

Automation and Quality Metrics

Software Test Automation, Skills needed for Automation, Scope of Automation, Design and Architecture for Automation, Requirements for a Test Tool, Challenges in Automation Tracking the Bug, Debugging. Testing Software System Security - Six-Sigma, TQM - Complexity Metrics and Models, Quality Management Metrics, Availability Metrics, Defect Removal Effectiveness, FMEA, Quality Function Deployment, Taguchi Quality Loss Function, Cost of Quality.

[8 hrs] (CO 3)

Module 4:

Quality Assurance tools and Models

SQA basics, Components of the Software Quality Assurance System, software quality in business context, planning for software quality assurance, product quality and process quality, software process models, 7 QC Tools and Modern Tools.

Models for Quality Assurance, ISO-9000 series, CMM, CMMI, Test Maturity Models, SPICE, Malcolm Baldrige Model- P-CMM **[8hrs] (CO4)**

Module 5:

Quality Assurance trends;

Software Process- PSP and TSP, OO Methodology, Clean-room software engineering, Defect Injection and prevention, Internal Auditing and Assessments, Inspections & Walkthroughs, Case Tools and their Affect on Software Quality.

[6hrs] (CO5)

Text Books:

1. Srinivasan Desikan, Gopalaswamy Ramesh,Software Testing: Principles and Practices Pearson.
2. 2. Daniel Galin, Software Quality Assurance: From Theory to Implementation, Pearson Addison Wesley.

Reference Books:

3. Aditya P. Mathur, Foundations of Software Testing, Pearson.
4. Paul Ammann, Jeff Offutt, Introduction to Software Testing, Cambridge University Press.
5. Paul C. Jorgensen, Software Testing: A Craftsman's Approach, Auerbach Publications.
6. William Perry, Effective Methods of Software Testing, Wiley Publishing, Third Edition.
7. Renu Rajani, Pradeep Oak, Software Testing – Effective Methods, Tools and Techniques, Tata McGraw Hill.

COURSE Outcomes: By the end of the course, students should be able to

1. Test the software by applying testing techniques to deliver a product free from bugs.

2. Investigate the scenario and to select the proper testing technique.

3. Explore the test automation concepts and tools and estimation of cost, schedule based

on standard metrics.

4. Understand how to detect, classify, prevent and remove defects.

5. Choose appropriate quality assurance models and develop quality.

Course Code: SE504-19	**Course Title:** Software testing& quality assurance Lab	**0L:0T:2P**	**1Credits**

Detailed

Contents:

1.To determine the nature of roots of a quadratic equations, its input is triple of +ve integers (say x,y,z) and values may be from interval[1,100] the program output may have one of the following:- [Not a Quadratic equations, Real roots, Imaginary roots, Equal roots] Perform BVA.

2. To determine the type of triangle. Its input is triple of +ve integers (say x,y,z) and the values may be from interval[1,100].The program output may be one of the following [Scalene, Isosceles, Equilateral, Not a Triangle].Perform BVA

3. Perform robust case testing on Problem No. 1.

4. Perform robust case testing on Problem No. 2.

5. Create a test plan document for any application (e.g. Library Management System)

6. Experiment: Study of Any Testing Tool (Win Runner)

7. Experiment: Study of Any Test Management Tool (QA Complete)

8. Experiment: Automate the Test cases using Test Automation tool(using QA Complete)

9. Experiment: Learn how to raise and report Bugs using Bug tracking tool (Bugzilla,Jira using QA Complete)

10. Experiment: Study of any open source testing tool (Web Performance Analyzer/O STA)

ELECTIVES- I

Course Code: BTCS 510-18	**Course Title:** Programming in Python	**3L:0T:0P**	**3 Credits**	**42 Hours**

Detailed Contents:

Module 1:

Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules Sequences - Strings, Lists, and Tuples, Mapping and Set Types.

[8hrs] (CO1)

Module 2:

FILES: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules, Related Modules

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, *Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, *Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules

Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules.

[10hrs] (CO1,2)

Module 3:

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules.

[8hrs] (CO 2,3)

Module 4:

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

WEB Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application Advanced CGI, Web (HTTP) Servers.
[10hrs] (CO 4,6)

Module 5:

Database Programming: Introduction, Python Database Application Programmer’s Interface (DB-API), Object Relational Managers (ORMs), Related Modules. **[6 hrs] (CO5)**

Text Books:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

Course Outcomes:

The students should be able to:

CO1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
CO2: Demonstrate proficiency in handling Strings and File Systems.
CO3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
CO4: Interpret the concepts of Object-Oriented Programming as used in Python.
CO5: Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Course Code: BTCS 513-18	**Course Title:** Programming in Python Lab	**0L:0T:2P**	**1 Credits**

Prerequisites: Students should install Python.

List of Experiments:

Task 1: Write a program to demonstrate different number data types in Python.
Task 2: Write a program to perform different Arithmetic Operations on numbers in Python.
Task 3: Write a program to create, concatenate and print a string and accessing sub-string from a given string.
Task 4: Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”
Task 5: Write a program to create, append, and remove lists in python.
Task 6: Write a program to demonstrate working with tuples in python.
Task 7: Write a program to demonstrate working with dictionaries in python.
Task 8: Write a python program to find largest of three numbers.
Task 9: Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]
Task 10: Write a Python program to construct the following pattern, using a nested for loop *

```
*
* *
* * *
* * * *
* * *
* *
*
*
```

Task 11: Write a Python script that prints prime numbers less than 20.
Task 12: Write a python program to find factorial of a number using Recursion.
Task 13: Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the

Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).

Task 14: Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

Task 15: Write a python program to define a module and import a specific function in that module to another program.

Task 16: Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

Task 17: Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.

Task 18: Write a Python class to convert an integer to a roman numeral.

Task 19: Write a Python class to implement pow(x, n)

Task 20: Write a Python class to reverse a string word by word.

----

cCourse Code: BTCS521-18	**Course Title:** Computational Biology	**3L:0T:0P**	**3 Credits**	**42 Hours**

Detailed Contents:

Module 1: Introduction

Nature and scope of life science: Branches of life sciences, Characteristics of life, Levels of Organization, Origin of life, Biochemical evolution- evolution of Proteins and Nucleotide. ***Cell Biology***: The cell as basic unit of life- Prokaryotic cell and Eukaryotic cell, Cell Structure and Function- cell membrane, cell organelles, Cell Division; Mitosis & Meiosis. ***Cell Energetics***: Laws of Thermodynamics, Photosynthesis, Anaerobic & aerobic respiration, Structure and function of mitochondria, respiratory pathways: Glycolysis, Kreb's Cycle, Electron transport chain.

[10hrs] (CO)

Module 2: More about RNA and DNA

Chromosome-Genome-Genes-Databases: Bio-molecules- DNA, RNA, Protein and amino acids, Chargaff's Rules, GC content.

Central Dogma: Replication, Transcription, Translation, Post transcriptional & post translational modifications, RNA processing, RNA splicing and RNA editing. Sense/coding and anti-sense/template strands, Genetic code. Introduction to DNA and Protein sequencing.

[10hrs] (CO)

Module 3: Proteins

Proteins and Databases: Protein structure and function, Protein Primary structure, Amino acid residues, Secondary, Tertiary, Quaternary Structure of Protein, Protein sequence databases- SwissProt/ TrEMBL, PIR, Sequence motif databases -Pfam, PROSITE, Protein structure databases.

[8hrs] (CO)

Module 4: Computation and Biology

Molecular computational biology: Gene prediction, sequencing genomes, similarity search, restriction mapping, ***Sequence Analysis***: Principles and its uses, Hidden Markov models for sequence analysis. Introduction of Markov Chain and Hidden Markov models. Forward backward algorithm, Viterbi and Baum-Welch algorithms,

[14hrs] (CO)

Course Outcomes:

The student will be able to:

CO1: Understand the basic of cell structure, divisions involved in reproduction of a cell, and its generic functionality;

CO2: Recognize the base line elements of a RNA and DNA; including fundamental behind their complex structure;

CO3: Comprehend primary structure of the protein and various related data-sets.

CO4: Demonstrate the concept of gene sequence alignment and simulate various related algorithms for the same.

Text books

1. Pevzner, P. A., Computational Molecular Biology, PHI Learning Pvt. Ltd, ISBN-978-81-203-2550-0.
2. Ghosh, Z. and Mallick, B., Bioinformatics Principles and Applications (2008) Oxford University Press ISBN 9780195692303
3. Mount, D. W., Bioinformatics – sequence and genome analysis.

Reference Books

1. Devasena, T. (2012). Cell Biology. Published by Oxford University Press.
2. Fall, C.P., Marland, E.S., Wagner, J.M., Tyson, J.J.(2002). Computational Cell Biology. Springer
3. Becker, W. M., Kleinsmith, L. J., Hardin, J., & Raasch, J. (2003). The world of the cell (Vol. 6). San Francisco: Benjamin Cummings.
4. Rastogi, S. C. (2005). Cell biology. New Age International.
5. Reece, J. B., Taylor, M. R., Simon, E. J., & Dickey, J. (2009). Biology: concepts & connections (Vol. 3, p. 2). Pearson/Benjamin Cummings.

Course Code: BTCS523-18	**Course Title:** Computational Biology Lab	**0L:0T:2P**	**1 Credits**	**2 Hours/ week**

List of Experiments:

Task 1: Introduction of Bio Python, Various Packages and its Installation.

Task 2,3: Parsing sequence file formats
Sequences and Alphabets
Sequences act like strings

- Slicing a sequence
- Turning Seq objects into strings
- Concatenating or adding sequences
- Changing case
- Nucleotide sequences and (reverse) complements
- Transcription
- Translation

Task 4,5: Sequence annotation objects

- The SeqRecord object
- Creating a SeqRecord
  - SeqRecord objects from scratch
  - SeqRecord objects from FASTA files
  - SeqRecord objects from GenBank files
- Feature, location and position objects
  - SeqFeature objects
  - Positions and locations
  - Sequence described by a feature or location

Task 6,7,8: BLAST

- Running BLAST over the Internet
- Running BLAST locally
  - Introduction
  - Standalone NCBI BLAST+
  - Other versions of BLAST
- Parsing BLAST output
- The BLAST record class
- Dealing with PSI-BLAST
- Dealing with RPS-BLAST

BLAST and other sequence search tools

- The SearchIO object model
  - QueryResult
  - Hit
  - HSP
  - HSPFragment
- A note about standards and conventions
- Reading search output files
- Dealing with large search output files with indexing
- Writing and converting search output files

Task 9,10: Multiple Sequence Alignment objects

- Parsing or Reading Sequence Alignments
  - Single Alignments
  - Multiple Alignments
  - Ambiguous Alignments

Writing Alignments

Converting between sequence alignment file formats

Getting your alignment objects as formatted strings

Manipulating Alignments

Slicing alignments

Alignments as arrays

Task 11,12,13: Sequence motif analysis using Bio.motifs

Motif objects

Creating a motif from instances

Creating a sequence logo

Reading motifs

JASPAR

MEME

TRANSFAC

Writing motifs

Position-Weight Matrices

Quick Reference:
http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc106
https://biopython.readthedocs.io/en/latest/Tutorial/chapter_seq_objects.html

Course Code: BTCS 515-18	**Course Title:** Computer Graphics	**3L:0T:0P**	**3 Credits**	**45 Hours**

Detailed Contents:

Module 1:

Overview of Computer Graphics: Basics of Computer Graphics, Applications, Video Display devices, Raster–Scan displays, Random–Scan displays, Color CRT Monitors, Flat–Panel Displays; Video Controller, Display Processor, Common Graphic Input and Output devices, Graphic File Formats, Graphics Software's.

[6hrs] (CO1)

Module 2:

Output Primitives: Line Drawing, DDA, Bresenham Line Algorithm; Mid-Point Line Algorithm, Bresenham Circle Algorithm, Midpoint Circle drawing algorithms; Midpoint Ellipse Algorithm; Flood and Boundary Filling.

[6hrs] (CO1)

Module 3:

Two-Dimensional Geometric Transformation: Translation, Rotation, Scaling, Reflection, Shearing, Matrix representations; Composite transformations.

[6hrs] (CO1,2)

Module 4:

Two-Dimensional Viewing: Viewing coordinate reference frame; Window to Viewport coordinate transformation. Point Clipping, Line Clipping, text Clipping; Cohen–Sutherland and Liang–Barskey Algorithms for line clipping; Sutherland–Hodgeman algorithm for polygon clipping.

[6hrs](CO2)

Module 5:

Three Dimensional Transformations & Viewing: Translation, Rotation, Scaling, Reflection and composite transformations. Parallel and Perspective Projections, Viewing Transformation: View Plan, View Volumes and Clipping.

[6hrs] (CO2)

Module 6:

3 D Graphics and Visibility: Plane projections and its types, Vanishing points, Specification of a 3D view. Image and object precision, Hidden edge/surface removal or visible edge/surface determination techniques; z buffer algorithms, Depth sort algorithm, Scan line algorithm and Floating horizon technique.

[6hrs] (CO2,3)

Module 7:

Color Models: Properties of Light, Intuitive Color Concepts, concepts of chromaticity, RGB Color Model, CMY Color Model, HLS and HSV Color Models, Conversion between RGB and CMY color Models, Conversion between HSV and RGB color models, Color Selection and Applications.

[6hrs] (CO2,3)

Module 8:

Animation: Graphics Design of Animation sequences, General Computer Animation Functions Introduction to Rendering, Raytracing, Antialiasing, Fractals, Gourard and Phong shading.

[3hrs] (CO3)

Reference Books:

1. D. Hearn and M.P. Baker, Computer Graphics: C version, 2nd Edition, PHI, 2004.
2. D.F. Rogers, Mathematical Elements for Graphics, 2nd Edition., McGraw Hill, 2004.
3. J.D. Foley et al, Computer Graphics, Principles and Practices, 2nd Edition, Addison Wasley, 2004.
4. Roy A. Plastock, Gordon Kalley, Computer Graphics, Schaum's Outline Series, 1986.

Course Outcomes: The students shall be able to:

CO1: Understand about fundamentals of Graphics to enable them to design animated scenes for virtual object creations.

CO2: Make the student present the content graphically.

CO3: Work in computer aided design for content presentation for better analogy data with pictorial representation

----

Course Code: BTCS 518-18	**Course Title:** Computer Graphics Lab	**0L:0T:4P**	**2 Credits**	**2 Hours/ week**

List of Experiments:

Task 1: WAP to draw different geometric structures using different functions.
Task 2: Implement DDA line generating algorithm.
Task 3: Implement Bresenham's line generating algorithm.
Task 4: Implement Mid-point circle line generating algorithm.
Task 5: Implementation of Bresenham's circle drawing algorithm.
Task 6: Implementation of mid-point circle generating Algorithm.
Task 7: Implementation of ellipse generating Algorithm.
Task 8: WAP of color filling the polygon using Boundary fill and Flood fill algorithm.
Task 9: To translate an object with translation parameters in X and Y directions.
Task 10: To scale an object with scaling factors along X and Y directions.
Task 11: Program of line clipping using Cohen-Sutherland algorithm.
Task 12: To perform composite transformations of an object.
Task 13: To perform the reflection of an object about major.

Course Code: BTCS 520-18	**Course Title:** Web Technologies	**3L:0T:0P**	**3 Credits**	**42 Hours**

Detailed Contents:

Module 1:

Introduction: History and evolution of Internet protocols, Internet addressing, Internet Service Provider (ISP), Introduction to WWW, DNS, URL, HTTP, HTTPS, SSL, Web browsers, Cookies, Web servers, Proxy servers, Web applications. Website design principles, planning the site and navigation. [6 hrs][CO1]

Module 2:

HTML and DHTML: Introduction to HTML and DHTML, History of HTML, Structure of HTML Document: Text Basics, Structure of HTML Document: Images, Multimedia, Links, Audio, Video, Table and Forms, Document Layout, HTML vs. DHTML, Meta tags, Browser architecture and Website structure. Overview and features of HTML5. [7 hrs][CO2]

Module 3:

Style Sheets: Need for CSS, Introduction to CSS, Basic syntax and structure, Types of CSS – Inline, Internal and External CSS style sheets.CSS Properties - Background images, Colors and properties,Text Formatting, Margin, Padding, Positioning etc., Overview and features of CSS3.
[7 hrs][CO3]

Module 4:

Java Script: Introduction, JavaScript's history and versions, Basic syntax, Variables, Data types, Statements, Operators, Functions, Arrays, Objects, dialog boxes, JavaScript DOM.
[7 hrs][CO4]

Module 5:

PHP and MySQL: Introduction and basic syntax of PHP, Data types, Variables,Decision and looping with examples, String, Functions, Array, Form processing, Cookies and Sessions, E-mail, PHP-MySQL: Connection to server.
[7 hrs][CO5]

Module 6:

Ajax and JSON: AJAX Introduction, AJAX Components, Handling Dynamic HTML with Ajax, Advantages & disadvantages, HTTP request, XMLHttpRequest Server Response.

JSON– Syntax, Schema, Data types, Objects, Reading and writing JSON on client and server. Using JSON in AJAX applications. [8 hrs][CO6]

Students shall be able to:

CO1. Understand and apply the knowledge of web technology stack to deploy various web services.
CO2. Analyze and evaluate web technology components for formulating web related problems.
CO3. Design and develop interactive client server internet application that accommodates user specific requirements and constraint analysis.
CO4. Program latest web technologies and tools by creating dynamic pages with an understanding of functions and objects.
CO5. Apply advance concepts of web interface and database to build web projects in multidisciplinary environments.
CO6. Demonstrate the use of advance technologies in dynamic websites to provide performance efficiency and reliability for customer satisfaction.

<u>Text Books</u>:

1. Jeffrey C. Jackson, “Web Technologies: A Computer Science Perspective”, PearsonEducation
2. Rajkamal, “Internet and Web Technology”, Tata McGraw Hill
3. Ray Rischpater, “JavaScript JSON Cookbook”, Packt Publishing.
4. Ivan Bayross, “Web Enabled Commercial Application Development using HTML, DHTML JavaScript, Perl, CGI”, BPB Publications.
5. Peter Moulding, “PHP Black Book”, Coriolis.

Course Code: BTCS 522-18	**Course Title:** Web Technologies Lab	**0L:0T:2P**	**1 credits**	**2 Hours/ week**

List of Experiments:

1. Configuration and administration Apache Web Server.
2. Develop an HTML page to demonstrate the use of basic HTML tags, Link to different HTML page and also link within a page, insertion of images and creation of tables.
3. Develop a registration form by using various form elements like input box, text area, radio buttons, check boxes etc.
4. Design an HTML page by using the concept of internal, inline, external style sheets.
5. Create an HTML file to implement the styles related to text, fonts, links using cascading style sheets
6. Create an HTML file to implement the concept of document object model using JavaScript
7. Create an HTML page including JavaScript that takes a given set of integer numbers and shows them after sorting in descending order.
8. Write an HTML page including any required JavaScript that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show “out of range” and if it is not a number, it should show “not a number” message in the result box.
9. Create a PHP file to print any text using variable.
10. Demonstrate the use of Loops and arrays in PHP
11. Create a PHP file using GET and POST methods.
12. A simple calculator web application that takes two numbers and an operator (+, -, /, * and %) from an HTML page and returns the result page with the operation performed on the operands.
13. Implement login page contains the user name and the password of the user to authenticate with Session using PHP and MySQL, also implement this with the help of PHP-Ajax.
14. A web application for implementation:
    a. The user is first served a login page which takes user’s name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
    b. If name and password matches, serves a welcome page with user’s full name.
    c. If name matches and password doesn’t match, then serves “password mismatch” page
    d. If name is not found in the database, serves a registration page, where user’s full name is asked and on submitting the full name, it stores, the login name, password and full name in the database (hint: use session for storing the submitted login name and password)
15. Demonstrate the use of Ajax and JSON Technologies in programming examples.

16. Demonstrate the use of web site designing tools such as Joomla, WordPress.
17. Implement at least one minor project using different technologies mentioned in theory of the subject.

Sixth Semester

Course Code: BTCS601-18	Course Title : Compiler Design	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: Unit I Introduction to Compilers:

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

[8 hrs., CO 1]

Unit II :Syntax Analysis:

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

[8 hrs., CO 2]

Unit III : Intermediate Code Generation:

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

[8 hrs., CO 3]

Unit IV: Run-Time Environment and Code Generation:

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

[6 hrs., CO 4]

Unit V: Code Optimization:

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

[6 hrs., CO 5]

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Build concepts on lexical analysis.

CO2: Understand strategies of syntax analysis.

CO3: Learn techniques of Intermediate code generation.

CO4: Undestand code design issues and design code generator.

CO5: Design and develop optimized codes.

Suggested Readings/ Books:

1. A.V. Aho, Monica, R.Sethi, J.D.Ullman, “Compilers, Principles, Techniques and Tools”, Second Edition, Pearson Education/Addison Wesley, 2009.
2. Andrew W. Appel, “Modern Compiler Implementation in Java”, Second Edition, 2009.
3. J.P. Tremblay and P.G. Sorrenson, “The Theory and Practice of Compiler Writing”, McGraw Hill, 1985.

Course Code: BTCS604-18	Course Title: Compiler Design Lab	L:0;T:0; 2P	1Credits

Detailed Contents:

Sr. No.	No. List of Experiments
1	Design a lexical analyser for given language and the lexical analyser should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C language.
2	Write a C program to identify whether a given line is a comment or not.
3	Write a C program to recognize strings under 'a', 'a*b+', 'abb'.
4	Write a C program to test whether a given identifier is valid or not.
5	Write a C program to simulate lexical analyzer for validating operators.
6	Implement the lexical analyzer using JLex, flex or other lexical analyzer generating tools.
7	Write a C program for implementing the functionalities of predictive parser for the mini language specified in Note 1.
8	a) Write a C program for constructing of LL (1) parsing. b) Write a C program for constructing recursive descent parsing.
9	Write a C program to implement LALR parsing.
10	a) Write a C program to implement operator precedence parsing. b) Write a C program to implement Program semantic rules to calculate the expression that takes an expression with digits, + and * and computes the value.
11	Convert the BNF rules into YACC form and write code to generate abstract syntax tree for the mini language specified in Note 1.
12	Write a C program to generate machine code from abstract syntax tree generated by the parser. The instruction set specified in Note 2 may be considered as the target code.

Lab Outcomes: At the end of this course student will:

1. Design Lexical analyzer for given language using C and LEX tools.
2. Design and convert BNF rules into YACC form to generate various parsers.
3. Generate machine code from the intermediate code forms
4. Implement Symbol table

Course Code:BTCS602-18	**Course Title : Artificial Intelligence**	**3L:0T:0P**	**3Credits**

Detailed Contents:

UNIT 1: Introduction (3 Hours)

Concept of AI, history, current status, scope, agents, environments, Problem Formulations, Review of tree and graph structures, State space representation, Search graph and Search tree.

[**8hrs] (CO 1)**

UNIT 2: Search Algorithms

Random search, Search with closed and open list, Depth first and Breadth first search, Heuristic search, Best first search, A* algorithm, Game Search.

[9hrs] (CO 2)

UNIT 3: Probabilistic Reasoning

Probability, conditional probability, Bayes Rule, Bayesian Networks- representation, construction and inference, temporal model, hidden Markov model.

[6hrs] (CO 3)

UNIT 4 Markov Decision process

MDP formulation, utility theory, utility functions, value iteration, policy iteration and partially observable MDPs. **[6hrs] (CO 4)**

UNIT 5 Reinforcement Learning

Passive reinforcement learning, direct utility estimation, adaptive dynamic programming, temporal difference learning, active reinforcement learning- Q learning**.**

[6hrs] (CO 5)

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Build intelligent agents for search and games

CO2: Solve AI problems by learning various algorithms and strategies

CO3: Understand probability as a tool to handle uncertainity

CO4: Learning optimization and inference algorithms for model learning

CO5: Design and develop programs for an reinforcement agent to learn and act in a structured environment

Suggested Readings/ Books:

1.Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach" , 3rd Edition, Prentice Hall

2. Elaine Rich and Kevin Knight, "Artificial Intelligence", Tata McGraw Hill

3. Trivedi, M.C., "A Classical Approach to Artifical Intelligence", Khanna Publishing House, Delhi.

4. Saroj Kaushik, "Artificial Intelligence", Cengage Learning India,

5. David Poole and Alan Mackworth, "Artificial Intelligence: Foundations for Computational Agents", Cambridge University Press 2010

Course Code: BTCS 605-18	**Course Title Artificial Intelligence Lab**	**L:0;T:0;P:2**	**1 Credits**

Detailed List of Tasks:

1.Write a programme to conduct uninformed and informed search.

2. Write a programme to conduct game search.

3. Write a programme to construct a Bayesian network from given data.

4. Write a programme to infer from the Bayesian network.

5. Write a programme to run value and policy iteration in a grid world.

6. Write a programme to do reinforcement learning in a grid world

Lab Outcomes: At the end of the course, the students are able to:

1. Explain artificial intelligence, its characteristics and its application areas.
2. Formulate real-world problems as state space problems, optimization problems or constraint satisfaction problems.

3. Select and apply appropriate algorithms and AI techniques to solve complex problems.
4. Design and develop an expert system by using appropriate tools and techniques.

Elective –II

Course Code: SE 601-19	Course Title: Object Oriented Analysis & Design	3L:0T:0P	3Credits

Detailed Contents:

Module 1:

Object Oriented Design and Modelling: Object Oriented Fundamentals, Objects and object classes, object oriented design process, importance of modelling, principles of modelling, object oriented modelling.

Introduction to UML: Conceptual model of UML, building blocks of UML, Mechanisms in UML, architecture, software development life cycle.

Module 2:

Basic Structural Modelling: Classes, relationships, common mechanisms, class and object diagrams.

Advanced structural Modelling: Advanced classes, advanced relationships, Interfaces types and roles, packages, instances and object diagrams.

Module 3:

Collaboration Diagrams and Sequence Diagrams: Terms, concepts and depicting a message in collaboration diagrams. Terms and concepts in sequence diagrams. Difference between collaboration and sequence. diagram. Depicting synchronous messages with/without priority call back mechanism.

Basic behavioural modelling: Interactions, use cases, Use Case Diagrams, Interaction Diagrams and activity diagrams.

Module 4:

Advanced behavioural modelling: Events and signals, state machines, process and threads, time and space, state chart diagrams.

Architectural Modelling: Terms, Concepts, examples, Modelling techniques for component diagrams and deployment diagrams.

Course Outcomes:

The student will be able to:

CO1: Explain OOAD concepts and various UML diagrams
CO2: Select an appropriate design pattern
CO3: Illustrate about domain models and conceptual classes
CO4: Compare and contrast various testing techniques
CO5: Construct projects using UML diagrams

Suggested Books

1. Grandy Booch, James Rumbough, Ivar Jacobson. ‘ The Unified Modelling Language User Guide. Pearson Edutaion 2002.
2. Ian Sommerville, ‘ Software Engineering Sixth Edition’ 2003.
3. Meilir Page Jones, ‘ Fundamentals of Object Oriented Design in UML’ , Addison Wesley, 2000

Course Code: SE 602-19	**Course Title:** Object Oriented Analysis & Design Lab	**0L:0T:2P**	**1 Credits**

List of Experiment:

Task 1: To develop a problem statement for the identified mini-project.
Task 2: Develop an IEEE standard Software Requirements Specification (SRS) document for the identified mini-project (systems) and the risk management and project plan (Gantt chart)
Task 3: Identify Use Cases and develop the Use Case model.
Task 4: Identify the business activities and develop a UML Activity diagram.
Task 5: Identity the conceptual classes and develop a domain model with a UML Class diagram.
Task 6: Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.
Task 7: Draw the State Chart diagram.
Task 8: Identify the User Interface, Domain objects, and Technical Services.
Task 9: Draw the partial layered, logical architecture diagram with UML package diagram notation.
Task 10: Implement the Technical services layer.
Task 11: Implement the Domain objects layer.
Task 12: Implement the User Interface layer.
Task 13: Draw Component and Deployment diagrams.

Lab Outcomes:
The student will be able to:
CO1: Perform OO analysis and design for a given problem specification.
CO2: Identify and map basic software requirements in UML mapping.
CO3: Improve the software quality using design patterns and explain the rationale behind applying specific design patterns
CO4: Test the compliance of the software with the SRS.

Suggested Software Tools:
ArgoUML, Eclipse IDE, Visual Paradigm, Visual case, and Rational Suite.

Course Code: BTCS614-18	Course Title: Software Project Management	3L:0T:0P	3 Credits

Detailed Contents:

MODULE 1: Introduction

Project Evaluation and Planning - Activities in Software Project Management, Overview of Project Planning, Stepwise planning, contract management, Software processes and process models.

[5hrs] (CO1)

MODULE 2: Cost Benefit Analysis

Cost Benefit Analysis, Cash Flow Forecasting, Cost-Benefit Evaluation Techniques, Risk Evaluation. Project costing, COCOMO 2, Staffing pattern, Effect of schedule compression, Putnam's equation, Capers Jones estimating rules of thumb.

[6hrs] (CO2)

MODULE 3: Project Scheduling

Project Sequencing and Scheduling Activities, Scheduling resources, Critical path analysis, Network Planning, Risk Management, Nature and Types of Risks, Managing Risks, Hazard Identification, Hazard Analysis, Risk Planning and Control, PERT and Monte Carlo Simulation techniques.

[8hrs] (CO3)

MODULE 4: Monitoring & Control

Monitoring and Control- Collecting Data, Visualizing Progress, Cost Monitoring, review techniques, project termination review, Earned Value analysis, Change Control, Software Configuration Management (SCM), Managing Contracts, Types of Contracts, Stages in Contract Placement, Typical Terms of a Contract, Contract Management and Acceptance.

[8hrs] (CO4)

MODULE 5: Quality Management

Quality Management and People Management- Introduction, Understanding Behavior, Organizational Behavior, Selecting the Right Person for The Job, Motivation, The Oldman – Hackman Job Characteristics Model, Working in Groups, Organization and team structures, Decision Making, Leadership, Organizational Structures, Stress, Health and Safety. ISO and CMMI models, Testing, and Software reliability, test automation, Overview of project management tools.

[9hrs] (CO5)

Course Outcomes:

After undergoing this course, the students will be able to:

- CO1: Explain project management in terms of the software development process
- CO2: Estimate project cost and perform cost-benefit evaluation among projects
- CO3: Apply the concepts of project scheduling and risk management.
- CO4: Explain Software configuration management and the concepts of contract management.
- CO5: Apply quality models in software projects for maintaining software quality and reliability

Suggested Readings/Books:

1. Bob Hughes, Mike Cotterell, “Software Project Management”, Tata McGraw Hill. (2009)

2. Royce, “Software Project Management”, Pearson Education. (2005).

3. Robert K. Wysocki, “Effective Software Project Management”, Wiley. (2006)

4. Ian Sommerville, Software Engineering, Seventh Edition, Pearson Education.

5. R.S. Pressman, Software Engineering: A Practitioner's Approach, Sixth Edition, Tata McGraw-Hill.

6. Kassem, Software Engineering, Cengage Learning

Course Code: BTCS 615-18	Course Title: Software Project Management Lab	L:0;T:0; P:2	1 Credits

Detailed List of Tasks:

Task 1: Introduction to MS Project

Task 2: Create a Project Plan

- Specify project name and start (or finish) date.
- Identify and define project tasks.
- Define duration for each project task.
- Define milestones in the plan
- Define dependency between tasks

Task 3: Create Project Plan contd.

- Define project calendar.
- Define project resources.
- Specify resource type and resource rates
- Assign resources against each task
- Baseline the project plan

Task 4: Execute and Monitor the Project Plan

- Update % Complete with current task status.
- Review the status of each task.
- Compare Planned vs Actual Status
- Review the status of Critical Path
- Review resources assignation status

Task 5: Generate Dashboard and Reports

- Dashboard
- Resource Reports
- Cost Reports
- Progress Reports

Suggested Tools – MS Project, Rational Team Concert

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Plan and manage projects.

CO2: Consolidate and communicate information about their project.

CO3: Create Gantt charts and PERT (Project Evaluation Review Technique) chart of their project

CO4: Manage resources, assignments, work allocation and generate reports to assess project status, project cost status and resource utilization.

CO5: Identify factors affecting the critical path of their project.

Course Code: BTCS 612-18	Course Title: Cloud Computing	3L:0T:0P	3Credits

Detailed Contents:

UNIT1: **Introduction** : Definition of cloud, characteristics of cloud, historical developments & challenges ahead, the vision of cloud computing, Driving factors towards cloud, Comparing grid with utility computing, cloud computing and other computing systems, types of workload patterns for the cloud, IT as a service, Applications of cloud computing.

[**8hrs] (CO1)**

UNIT2: Cloud computing concepts: Introduction to virtualization techniques, Characteristics of virtualization, Pros and Cons of virtualization Technology, Hypervisors, Types of hypervisors, Multitenancy, Application programming interfaces (API), Elasticity and scalability. **[9hrs] (CO2)**

UNIT 3: Cloud service models: Cloud service models, Infrastructure as a service (IaaS) architecture-details and example, Platform as a service (PaaS) architecture- details and example, Software as a service (SaaS) architecture-- details and example, Comparison of cloud service delivery models. **[6hrs] (CO3)**

UNIT 4: Cloud deployment models: Introduction to cloud deployment models, Public clouds, Private clouds, Hybrid clouds, Community clouds, Migration paths for cloud, Selection criteria for cloud deployment. **[6hrs] (CO4)**

UNIT 5: Security in cloud computing: Understanding security risks, Principal security dangers to cloud computing, Internal security breaches, User account and service hijacking, measures to reduce cloud security breaches
Case Studies: Comparison of existing Cloud platforms /Web Services.

[6hrs] (CO5)

Course Outcomes:

After undergoing this course, the students will be able to:
CO1: Understand the core concepts of the cloud computing paradigm
CO2: Understanding importance of virtualization along with their technologies
CO3: Analyze various cloud computing service and deployment models and apply them to solve problems on the cloud.
CO4: Implementation of various security strategies for different cloud platform

Suggested Readings/ Books:

1. Raj Kumar Buyya, James Broberg, Andrezei M.Goscinski, “Cloud Computing: Principles and Paradigms”,Wiley 2011
2. Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, “Cloud Computing: A practical Approach”, McGraw Hill, 2010.
3. Barrie Sosinsky, “Cloud Computing Bible”, Wiley, 2011.
4. Judith Hurwitz, Robin Bllor, Marcia Kaufman, Fern Halper, “Cloud Computing for dummies”, 2009.

Reference Books

1. Rajkumar Buyya, Christian Vecchiola, S.Thamarai Selvi, "Mastering Cloud Computing" TMH 2013.
2. George Reese "Cloud Application Architectures", First Edition, O"Reilly Media 2009.
3. Dr. Kumar Saurabh "Cloud Computing" 2nd Edition, Wiley India 2012.

Course Code: BTCS 613-18	Course Title: Cloud Computing Lab	L:0;T:0; P:2	1 Credits

Detailed List of Tasks:

1. Install VirtualBox/VMware Workstation on different OS.
2. Install different operating systems in VMware.
3. Simulate a cloud scenario using simulator.
4. Implement scheduling algorithms.
5. To study cloud security management.
6. To study and implementation of identity management
7. Case Study - Amazon Web Services/Microsoft Azure/Google cloud services.

Suggested Tools –Matlab, Cloudsim

Lab Outcomes: The students, after the completion of the course, are expected to

1. Use the cloud tool kits.
2. Implement applications on the Cloud
3. To install cloud computing environments
4. To develop any one type of cloud
5. To explore future trends of cloud computing

Elective –III

Course Code: SE 603-19	Course Title: Advanced Software Engineering	3L:0T:0P	3Credits

Detailed Contents:

Module 1:

Life Cycle Models

- Waterfall Model
- Prototyping Models
- Incremental Development
- Spiral Model
- Rapid Application Development
- Component Model
- Agile Software Development
- Selection of appropriate development process

Module 2:

Formal Methods

- Basic concepts
- Mathematical Preliminaries
- Mathematical notations for Formal Specification
- Formal Specification Languages
- Z-Notations
- Ten commandments of formal methods
- Formal Methods- The Road Ahead

Module 3:

- Component-Based Software Engineering
  - Engineering of Component-based Systems
  - CBSE Process
  - Domain Engineering
  - Component-based Development
  - Classifying and Retrieving Components
  - Economics of CBSE
- Cleanroom Software Engineering
  - The Cleanroom Approach
  - Functional Specification
  - Cleanroom Design Cleanroom Testing

Module 4:

- Client/Server Software Engineering
  - The Structure of Client/Server Systems

    - Software Engineering for Client Server Systems
    - Analysis Modeling Issues
    - Design for Client Server Systems
    - Testing Issues.
- Web Engineering
    - The Attributes of Web-based Applications
    - WebE Process
    - Framework for WebE
    - Formulating/Analyzing Web-based Systems
    - Design for Web-based Applications
    - Testing Web-based Applications
    - Management Issues.
- Service Oriented Software Engineering
    - Services as Reusable Components
    - Service Engineering
    - Software Development with Service

Module 5:

- Reengineering
    - Business Process Reengineering
    - Software Reengineering
    - Reverse Reengineering
    - Restructuring, Forward Reengineering
    - Economics of Reengineering.
- Computer-Aided Software Engineering
    - Introduction
    - Building Blocks for CASE
    - Taxonomy of CASE Tools
    - Integrated CASE Environments
    - Integration Architecture
    - CASE Repository
    - Case Study of Tools like TCS Robot.

Course Outcomes:

CO1: Analyze the software life cycle models;
CO2: Identify the importance of the software development process;
CO3: Analyze the importance of CASE tools;
CO4: Design and develop correct and robust software products using advanced software engineering techniques;
CO5: Able to understand business requirements pertaining to software development.

Suggested Books

1. Roger S. Pressman, Software Engineering a Practitioners Approach, McGraw-Hill (2008).
2. J. Bowan, Formal Specification and Documentation using Z - A Case Study Approach, International Thomson Computer Press (2003).

3. Antoni Diller, Z., An Introduction to Formal Methods (second edition), Wiley, 2nd edition (1994)

References:

1. M. Dyer, The Cleanroon Approach to Quality Software Development, Wiley (1992).
2. Prowell, S., Trammell, C.J. and Poore, J.H, Cleanroom Software Engineering: Technology and Process, Addison-Wesley, Massachusetts (1999).
3. Allen, Frost, Yourdon, Component-Based Development for Enterprise Systems: Applying the Select Perspectives, Cambridge University Press (1998).
4. Zantinge and Adriaans, Managing Client/Server, Addison-Wesley (1996).

Course Code: SE 604-19	**Course Title:** Advanced Software Engineering Lab	**L:0;T:0; P:2**	**1 Credits**

Detailed List of Tasks:

1. Preparing Software Requirements Specifications
2. Identifying Domain Classes from the Problem Statements
3. Modeling UML Class Diagrams and Sequence diagrams
4. Modeling UML Use Case Diagrams and Capturing Use Case Scenarios
5. E-R Modeling
6. State chart and Activity Modeling
7. Modeling Data Flow Diagrams
8. Estimation of Project Metrics
9. Estimation of Test Coverage Metrics and Structural Complexity
10. Designing Test Suites
11. Preparing Final Project Report

Lab Outcomes: The students, after the completion of the course, are expected to

1. Design and develop real-time software projects with effective cost estimation and plan
2. Make feasibility study of a project
3. Specify the design and architectural style of the software products
4. Propose testing strategy for a given software

Course Code: SE 605-19	**Course Title: Service oriented Architecture**	**L:3;T:0; P:0**	**3 Credits**

Detailed Contents:

UNIT 1:

Introduction: Roots, Characteristics and Anatomy of SOA, Comparing SOA to client-server and distributed internet architectures, SOA component interrelation, Principles of service orientation
Service Oriented Architecture: Major components of the architecture SOAP, XML, HTTP, Cookies, WSDL, XML schema, UDDI, Interactions between components.

UNIT 2:

Introduction to Web services: Service descriptions, Messaging with SOAP, Message exchange Patterns, Coordination, Atomic Transactions, Business activities, Orchestration, Choreography, Service layer abstraction, Application Service Layer, Business Service Layer, Orchestration Service Layer

Analysis: Service oriented analysis, Business-centric SOA, Deriving business services- service modelling, Service Oriented Design, WSDL basics, SOAP basics, SOA composition guidelines, Entity-centric business service design, Application service design, Task centric business service design

UNIT 3:

SOA platform basics: SOA support in J2EE, Java API for XML-based web services (JAX-WS), Java architecture for XML binding (JAXB), Java API for XML Registries (JAXR), Java API for XML based RPC (JAX-RPC), Web Services Interoperability Technologies (WSIT), SOA support in .NET, Common Language Runtime, ASP.NET web forms, ASP.NET web services, Web Services Enhancements (WSE)

UNIT 4:

Security: WS-BPEL basics, WS-Coordination overview, WS-Choreography, WS-Policy, WSSecurity

Recommended Books

1. Achieving Service-Oriented Architecture: Applying an Enterprise Architecture Approach, Rick Sweeney, 2010
2. Service-Oriented Architecture: Concepts, Technology, and Design, Thomas Erl, Pearson Education, 2005

Course Code: SE 606-19	Course Title: Service oriented Architecture lab	L:0;T:0; P:2	1 Credits

Laboratory work: Installing and configuring web servers, building and implementing Web services using the latest tools (.NET, J2EE)

Course Code: SE 607-19	Course Title: Big Data Analytics	L:3 T:0 P:0	3 Credits

Detailed Contents:

Module I

Introduction: Big Data Overview, The rising and importance of data sciences, Big data analytics in industry verticals

Hadoop Architecture: Hadoop Architecture, Hadoop ecosystem components, Hadoop Storage: HDFS, Hadoop Processing: MapReduce Framework, Hadoop Server Roles

Module II

Data Analytics Lifecycle and methodology: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Communicating results, Deployment, Data exploration & preprocessing

Module III

Data Analytics - Theory & Methods: Measures and evaluation, Supervised learning, Linear/Logistic regression, o Decision trees, Naïve Bayes, Unsupervised learning, K-means clustering, Association rules, Unstructured Data Analytics, Technologies & tools, Text mining, Web mining

Module IV

The Endgame: Opertionalizing an Analytics project, Data Visualization Techniques, Creating final deliverables

Course Outcomes

1. Describe Big Data and its importance with its applications

2. Differentiate various big data technologies like Hadoop MapReduce, Pig, Hive, Hbase and No-SQL.

3. Apply tools and techniques to analyze Big Data.

4. Design a solution for a given problem using suitable Big Data Techniques

Text Books:

1. Hadoop: The Definitive Guide by Tom White

2. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph by David Loshin
3. Machine Learning by Tom M. Mitchell

Course Code: SE 608-19	**Course Title: Big Data Analytics Lab**	**L:0 T:0 P:2**	**1 Credits**

List of Experiments:

1. Hands-on with Map Reduce: Hadoop, Hive, MapR
2. Hands-on with NoSQL Databases: S3, Hadoop Distributed File System(HDFS)
3. Hands-on with Statistical Packages
4. Hands-on with Visual Data Analysis tools

Lab Outcomes:

CO1: Perform data gathering of large data from a range of data sources.

CO2: Critically analyse existing Big Data datasets and implementations, taking practicality, and usefulness metrics into consideration.

CO3: Select and apply suitable statistical measures and analyses techniques for data of various structure and content and present summary statistics

CO4: Employ advanced statistical analytical skills to test assumptions, and to generate and present new information and insights from large datasets

Seventh Semester

Course Code: BTCS 701-18	**Course Title :** Network Security and Cryptography	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: Introduction (3 Hours)

Introduction to Cryptography, Security Threats, Vulnerability, Active and Passive attacks, Security services and mechanism, Conventional Encryption Model, CIA model

[5hrs] (CO 1)

UNIT 2: Math Background

Modular Arithmetic, Euclidean and Extended Euclidean algorithm, Prime numbers, Fermat and Euler's Theorem

[5hrs] (CO 1)

UNIT 3: Cryptography

Dimensions of Cryptography, Classical Cryptographic Techniques Block Ciphers (DES, AES) : Feistal Cipher Structure, Simplifies DES, DES, Double and Triple DES, Block Cipher design Principles, AES, Modes of Operations Public-Key Cryptography : Principles Of Public-Key Cryptography, RSA Algorithm, Key Management, Diffie-Hellman Key Exchange, Elgamal Algorithm, Elliptic Curve Cryptography

[12hrs] (CO 2)

UNIT 4 Hash and MAC Algorithms

Authentication Requirement, Functions, Message Authentication Code, Hash Functions, Security Of Hash Functions And Macs, MD5 Message Digest Algorithm, Secure Hash Algorithm, Digital Signatures, Key Management : Key Distribution Techniques, Kerberos

[6hrs] (CO 3)

UNIT 5 Security in Networks

Threats in networks, Network Security Controls – Architecture, Encryption, Content Integrity, Strong Authentication, Access Controls, Wireless Security, Honeypots, Traffic flow security, Firewalls – Design and Types of Firewalls, Personal Firewalls, IDS, Email Security – PGP, S/MIME

[7hrs] (CO 4)

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Understand the fundamental principles of access control models and techniques, authentication and secure system design

CO2: Have a strong understanding of different cryptographic protocols and techniques and be able to use them.

CO3: Apply methods for authentication, access control, intrusion detection and prevention.

CO4: Identify and mitigate software security vulnerabilities in existing systems.

Suggested Readings/ Books:

1. Cryptography And Network Security Principles And Practice Fourth Edition, William Stallings, Pearson Education

2. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall PTR

3. Network Security Essentials: Applications and Standards, by William Stallings. Prentice Hall

4. Cryptography: Theory and Practice by Douglas R. Stinson, CRC press.

Course Code: BTCS -702-18	**Course Title Data Mining and : Data Warehousing**	**3L: 0T: 0P**	**Credits: 3**

<u>Detailed Contents:</u>

UNIT 1:

Data Warehousing Introduction: design guidelines for data warehouse implementation, Multidimensional Models; OLAP- introduction, Characteristics, Architecture, Multidimensional view Efficient processing of OLAP Queries, OLAP server Architecture ROLAP versus MOLAP Versus HOLAP and data cube, Data cube operations, data cube computation.

Data mining: What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Pre-processing, Measures of Similarity and Dissimilarity

[10hrs]

UNIT 2:

Data mining: Introduction, association rules mining, Naive algorithm, Apriori algorithm, direct hashing and pruning (DHP), Dynamic Item set counting (DIC), Mining frequent pattern without candidate generation (FP, growth), performance evaluation of algorithms

Classification: Introduction, decision tree, tree induction algorithms – split algorithm based on information theory, split algorithm based on Gini index; naïve Bayes method; estimating predictive accuracy of classification method

[10 hrs]

UNIT 3:

Cluster analysis: Introduction, partition methods, hierarchical methods, density based methods, dealing with large databases, cluster software

Search engines: Characteristics of Search engines, Search Engine Functionality, Search Engine Architecture, Ranking of web pages, The search engine history, Enterprise Search, Enterprise Search Engine Software.

[10 hrs]

UNIT 4:

Web data mining: Web Terminology and Characteristics, Locality and Hierarchy in the web, Web Content Mining, Web Usage Mining, Web Structure Mining, Web mining Software.

[8 hrs]

Suggested Readings / Books:

1. Carlo Vercellis, Business Intelligence: Data mining and Optimization for Decision Making, WILEY.
2. Han J., Kamber M. and Pei J. , b Data mining concepts and techniques, Morgan Kaufmann Publishers (2011) 3rd ed.
3. Pudi V., Krishana P.R., Data Mining, Oxford University press, (2009) 1st ed.
4. Adriaans P., Zantinge D., Data mining, Pearsoneducation press (1996), 1st ed.
5. Pooniah P. , Data Warehousing Fundamentals, Willey interscience Publication, (2001), 1st ed.

ELECTIVE IV

Course Code: BTCS 618-18	Course Title : Machine Learning	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: **Introduction:** Well-Posed learning problems, Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning, Unsupervised learning and Reinforcement learning.

[4hrs] (CO 1)

UNIT 2: Data Pre-processing: Need of Data Pre-processing, Data Pre-processing Methods: Data Cleaning, Data Integration, Data Transformation, Data Reduction; Feature Scaling (Normalization and Standardization), Splitting dataset into Training and Testing set.

[4hrs] (CO 2)

UNIT 3: Regression: Need and Applications of Regression, Simple Linear Regression, Multiple Linear Regression and Polynomial Regression, Evaluating Regression Models Performance (RMSE, Mean Absolute Error, Correlation, RSquare, Accuracy with acceptable error, scatter plot, *etc*.)

[6hrs] (CO 3)

UNIT 4 Classification: Need and Applications of Classification, Logistic Regression, Decision tree, Tree induction algorithm – split algorithm based on information theory, split algorithm based on Gini index; Random forest classification, Naïve Bayes algorithm; K-Nearest Neighbours (K-NN), Support Vector Machine (SVM), Evaluating Classification Models Performance (Sensitivity, Specificity, Precision, Recall, *etc*.). **Clustering**: Need and Applications of Clustering, Partitioned methods, Hierarchical methods, Density-based methods.

[12hrs] (CO 4)

UNIT 5 Association Rules Learning: Need and Application of Association Rules Learning, Basic concepts of Association Rule Mining, Naïve algorithm, Apriori algorithm. **Artificial Neural Network:** Need and Application of Artificial Neural Network, Neural network representation and working, Activation Functions. **Genetic Algorithms:** Basic concepts, Gene Representation and Fitness Function, Selection, Recombination, Mutation and Elitism.

[14hrs] (CO 5)

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Analyse methods and theories in the field of machine learning

CO2: Analyse and extract features of complex datasets

CO3: Deploy techniques to comment for the Regression

CO4: Comprehend and apply different classification and clustering techniques

CO5: Understand the concept of Neural Networks and Genetic Algorithm

Suggested Readings/ Books:

Text Books:

1. Mitchell M., T., Machine Learning, McGraw Hill (1997) 1stEdition.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2014) 3rdEdition.
3. Vijayvargia Abhishek, Machine Learning with Python, BPB Publication (2018)

Reference Books:

1. Bishop M., C., Pattern Recognition and Machine Learning, Springer-Verlag (2011) 2ndEdition.
2. Michie D., Spiegelhalter J. D., Taylor C. C., Campbell, J., Machine Learning, Neural and Statistical Classification. Overseas Press (1994).

………

Course Code: BTCS619-18	**Course Title: machine Learning Lab**	**L:0;T:0; P:2**	**1Credits**

Detailed List of Tasks:

1. Implement data pre-processing
2. Deploy Simple Linear Regression
3. Simulate Multiple Linear Regression
4. Implement Decision Tree
5. Deploy Random forest classification
6. Simulate Naïve Bayes algorithm
7. Implement K-Nearest Neighbors (K-NN), k-Means
8. Deploy Support Vector Machine, Apriori algorithm
9. Simulate Artificial Neural Network
10. Implement the Genetic Algorithm code

Suggested Tools Python/R/MATLAB

Course Code: BTCS620-18	Course Title: Mobile Application Development	L:3; T:0; P:0	3Credits

Details of course:

Unit-1

Introduction to Android: The Android Developing environment, Android SDK, Introduction to Open Handset Alliance, Development Framework, Application Fundamentals; Device Compatibility, System permissions, Understanding Anatomy of Android Application, Android Development Tools

6 hrs. ,CO 1

Unit-II

Getting started with Mobility: Mobility Landscape, Mobile Platforms, Mobile apps development, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Setting up the mobile apps development environment with emulator

6 hrs., CO1

Unit-III

Building block of Mobile apps: App user Interface Designing, Layout, User Interface elements, VUIs and Mobile Apps, Text to Speech Techniques, Designing the Right UI, Activity states and lifecycle, Interaction among activities

6 hrs., CO 2

Unit-IV

Sprucing up Mobile apps: App functionality beyond user interface- Threads, sync task, Services-states and life cycle, Notifications, Broadcast receivers, Telephony and SMS APIs Native data handling: on device file I/O, shared preferences, mobile databases such as SQLite, Working with a content provider

8 hrs., CO 3,4

Unit-V

Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User

Graphics and Multimedia: Performance and Multithreading, Graphics and UI Performance, Android Graphics, Mobile Agents and Peer-to-Peer Architecture, Android Multimedia

8 hrs., CO 4,5

Unit-VI

Platforms and Additional Issues: Development Process, Architecture, Design, Technology Selection, Testing, Security and Hacking, Active Transactions, More on Security

8hrs., CO 5

Course Outcomes:

CO 1: Describe those aspects of mobile programming that make it unique from programming for other platforms,
CO 2: Critique mobile applications on their design pros and cons,
CO 3: Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces,
CO 4: Program mobile applications for the Android operating system that use basic and advanced phone features, and
CO 5: Deploy applications to the Android marketplace for distribution

References:

1. Rick Rogers, John Lombardo, Meike Blake, "Android application development", Ist Edition, O'Reilly, 2010
2. Lauren Darcey and Shane Conder,"Android Wireless Application Development", 2nd ed. Pearson Education, 2011
3. Wei-Meng Lee , Beginning Android 4 development ,2012 by John Wiley & Sons
4. Jeff Mewherter, Scott Gowell,Wrox Publisher,"Professional Mobile Application Development", Ist Edition, 2012
5. Reto Meier, "Professional Android 4 Application Development", Wrox, 2012

………..

Course Code: BTCS621-18	**Course Title: Mobile Application Development Lab**	**L:0; T:0; P:2**	**1Credits**

LIST OF PRACTICALS

1. Introduction to Android platform. Introduction to the tools used in the lab. Create a simple application

2. Understand the app idea and design user interface/wireframes of mobile app

3. Set up mobile app development environment

4. Write a program using activity class to show different events.

5. Write a program to convert text to speech.

6. Develop and debug mobile app components – User interface, services, notifications, broadcast receivers, data components

7. Using emulator to deploy and run mobile apps

8. Testing mobile app- unit testing, black box testing and test automation

Course Code: BTCS 704-18	Course Title : Deep Learning	3L:0T:0P	3Credits

Detailed Contents:

UNIT 1: **Machine Learning Basics:** Learning, Under-fitting, Overfitting, Estimators, Bias, Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning, Unsupervised Learning and Stochastic Gradient Decent.

[4hrs] (CO 1)

UNIT 2: Deep Feedforward Network: Feed-forward Networks, Gradient-based Learning, Hidden Units, Architecture Design, Computational Graphs, Back-Propagation, Regularization, Parameter Penalties, Data Augmentation, Multi-task Learning, Bagging, Dropout and Adversarial Training and Optimization.

[4hrs] (CO 2)

UNIT 3: Convolution Networks: Convolution Operation, Pooling, Basic Convolution Function, Convolution Algorithm, Unsupervised Features and Neuroscientific for convolution Network.

[6hrs] (CO 3)

UNIT 4: Sequence Modelling: Recurrent Neural Networks (RNNs), Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Network, Recursive Neural Networks and Echo State networks.

[12hrs] (CO 4)

UNIT 5: Deep Generative Models: Boltzmann Machines, Restricted Boltzmann Machines, Deep Belief Networks, Deep Boltzmann Machines, Sigmoid Belief Networks, Directed Generative Net, Drawing Samples from Auto –encoders.

[14hrs] (CO 5)

Course Outcomes:

After undergoing this course, the students will be able to:

CO1: Comprehend the advancements in learning techniques

CO2: Compare and explain various deep learning architectures and algorithms.

CO3: Demonstrate the applications of Convolution Networks

CO4: Apply Recurrent Network for Sequence Modelling

CO5: Deploy the Deep Generative Models

Suggested Readings/ Books:

Text Books:

1. *Goodfellow L., Bengio Y. and Courville A., Deep Learning, MIT Press (2016).*
2. *Patterson J. and Gibson A., Deep Learning: A Practitioner's Approach, O'Reilly (2017), 1st ed.*

Reference Books:

1. *Haykin S., Neural Network and Machine Learning, Prentice Hall Pearson (2009), 3rd ed.*
2. *Geron A., Hands-on Machine Learning with Sci-kit and TensorFlow, O'Reilly Media (2017)*

………

Course Code: BTCS 705-18	**Course Title: Deep Learning Lab**	**L:0;T:0; 2P:**	**Credits;1**

Detailed List of Tasks:

- Creating a basic network and analyze its performance
- Deploy the Confusion matrix and simulate for Overfitting
- Visualizing a neural network
- Demo: Object Detection with pre-trained RetinaNet with Keras
- Neural Recommender Systems with Explicit Feedback
- Backpropagation in Neural Networks using Numpy
- Neural Recommender Systems with Implicit Feedback and the Triplet Loss
- Fully Convolutional Neural Networks
- ConvNets for Classification and Localization
- Text Classification and Word Vectors
- Character Level Language Model (GPU required)

Suggested Tools Python/R/MATLAB

ELECTIVE V

Course Code: SE 701-19	**Course Name:** Software Metrics	L:3; T:0; P:0	3 Credits

Detailed Contents:

UNIT 1

Fundamentals of Measurement: Measurement: what is it and why do it?: Measurement in Software Engineering, Scope of Software Metrics, The Basics of measurement: The representational theory of measurement, Measurement and models, Measurement scales and scale types, meaningfulness in measurement

CO1

UNIT 2

A Goal-Based Framework For Software Measurement: Classifying software measures, Determining what to Measure, Applying the framework, Software measurement validation, Performing Software Measurement validation Empirical investigation: Principles of Empirical Studies, Planning Experiments, Planning case studies as quasi-experiments ,Relevant and Meaningful Studies

CO1,CO2

UNIT 3

Software Metrics Data Collection: Defining good data ,Data collection for incident reports, How to collect data, Reliability of data collection Procedures Analyzing software measurement data: Statistical distributions and hypothesis testing, Classical data analysis techniques, Examples of simple analysis techniques

CO2

UNIT 4

Measuring internal product attributes: Size Properties of Software Size, Code size, Design size, Requirements analysis and Specification size, Functional size measures and estimators, Applications of size measures Measuring internal product attributes: Structure: Aspects of Structural Measures, Control flow structure of program units, Design-level Attributes, Object-oriented Structural attributes and measures

CO3

UNIT 5

Measuring External Product Attributes: Modelling software quality, Measuring aspects of quality, Usability Measures, Maintainability measures, Security Measures Software Reliability: Measurement and Prediction: Basics of reliability theory, The software reliability problem, Parametric reliability growth models, Predictive accuracy

CO3,CO4

Course Objectives:

Upon successful completion of the course, the student will be able to
CO1: Understand various fundamentals of measurement and software metrics
CO2: Apply frame work and analysis techniques for software measurement.
CO3: Apply internal and external attributes of software product for effort estimation.
CO4: Apply reliability models for predicting software quality
CO5: Design Metrics for object-oriented systems.

Suggested Books:

1 Software Metrics A Rigorous and Practical Approach, Norman Fenton, James Bieman , Third Edition, 2014 2014

2 Software metrics, Norman E, Fenton and Shari Lawrence Pfleeger, International Thomson Computer Press, 1997

1997

3 Metric and models in software quality engineering, Stephen H.Kan, Second edition, 2002, Addison-Wesley Professional. 2002

4 Measuring the Software Process, William A. Florac and Areitor D. Carletow, 1995, Addison – Wesley 1995

5 Practical Software Metrics for Project Management and Process Improvement, Robert B.Grady, 1992, Prentice Hall. 1992

………

Course Code: SE 702-19	**Course Name:** Software Metrics Lab	L:0; T:0; P:2	1 Credits

Course Objectives:

The students should be able to describe and implement various ----------

Experiment Task List:

1. Develop requirements specification for a given problem.

2. Develop DFD model (level-0, level-1 DFD and Data dictionary) of the project.

3. Develop Structured design for the DFD model developed.

4. Develop UML Use case model for a problem.

5.Develop sequence diagram.

6. Develop Class ……

Course Outcomes: At the end of the course, students will be able to:

Lab Outcome

1. Can produce the requirements and use cases the client wants for the software being produced.

2. Participate in drawing up the project plan. The plan will include at least extent and work assessments of the project, the schedule, available resources, and risk management can model and specify the requirements of mid-range software and their architecture.

3. create and specify such a software design based on the requirement specification that the software can be implemented based on the design.

4. Can assess the extent and costs of a project with the help of several different assessment methods.

5. Can impart state-of-the-art knowledge on Software Metriics and UML in an interactive manner through the Web.

…….

Course Code: SE 703-19	**Course Name:** Software Verification & Validation Testing	L:3; T:0; P:0	3 Credits

Detailed Contents:

Unit 1:
Introduction: Terminology, evolving nature of area, Errors, Faults and Failures, Correctness and reliability, Testing and debugging, Static and dynamic testing, Exhaustive testing Theoretical foundations: impracticality of testing all data, impracticality of testing all paths, no absolute proof of correctness.

Unit 2:
Software Verification and Validation Approaches and their Applicability: Software technical reviews; Software testing: levels of testing - module, integration, system, regression; Testing techniques and their applicability-functional testing and analysis, structural testing and analysis, error-oriented testing and analysis, hybrid approaches, integration strategies, transaction flow analysis, stress analysis, failure analysis, concurrency analysis, performance analysis; Proof of correctness; simulation and prototyping; Requirement tracing.

Unit 3:
Test Generation: Test generations from requirements, Test generation pats, Data flow analysis, Finite State Machines models for flow analysis, Regular expressions based testing, Test Selection, Minimizations and Prioritization, Regression Testing.

Unit 4:
Program Mutation Testing: Introduction, Mutation and mutants, Mutation operators, Equivalent mutants, Fault detection using mutants, Types of mutants, Mutation operators for C and Java.

Text Books:
1. Jorgensen C. P., Software Testing: A Craftsman's Approach, CRC Press (2014), 4th ed.
2. Fisher S. M., Software Verification and Validation: An Engineering and Scientific Approach, Springer(2007).

Reference Books:
1. Mathur P. A., Foundations of Software Testing, Pearson (2013), 2nd ed.
2. Beizer B., Software Testing Techniques, Van Nostrand Reinhold (1983), 1sted.
3. Rakitin R. S., Software Verification and Validation for Practitioners and Managers, Artech House (2001), 2nd ed.

Course outcomes
After the completion of the course, the student will be able to:
CO1. Comprehend the theoretical foundations of verification, validation & testing.
CO2. Comprehend software testing levels, testing techniques and their applicability.
CO3. Generate test cases from software requirements, data flows and finite state machines.
CO4. Perform fault detection using mutants for operators of C and Java language.

Course Code: SE 704-19	**Course Name:** Software Verification & Validation Testing Lab	L:0; T:0; P:2	1 Credits

LIST OF EXPERIMENTS:
Laboratory Work: To Use various verification and validation testing tools and to apply these tools on few examples and case studies.

……..

Course Code: SE 705-19	**Course Name:** Secure Software Development	L:3; T:0; P:0	3 Credits

Course Objectives:
The students should be able to describe and implement various secure software development models. To demonstrate knowledge of the distinction between critical and non-critical systems and to enable students to manage a project including planning, scheduling, and risk assessment.

Detailed Contents:

1 Software security life cycle, Software quality attributes, Security requirement gathering principles and guidelines 5

2 Software Assurance and Software Security, Threats to software security, Sources of software insecurity, Benefits of Detecting Software Security 5

3 Properties of Secure Software, Influencing the security properties of software, Asserting and specifying the desired security properties 5

4 Requirements Engineering for secure software: Introduction, the SQUARE process Model, Requirements elicitation and prioritization 5

5 Secure Software Architecture and Design: Introduction, software security practices for architecture and design: architectural risk analysis, software security knowledge for architecture and design: security principles, security guidelines and attack patterns 5

6 Secure coding and Testing: Code analysis, Software Security testing, Security testing considerations throughout the SDLC 6

7 Security and Complexity: System Assembly Challenges: introduction, security failures, functional and attacker perspectives for security analysis, system complexity drivers and security 5

8 Secure Software design principles Static analysis techniques Security testing (black box and white box) 6

Course Outcomes: At the end of the course, students will be able to:

CO1: Explain the role of security throughout the Software Development Life Cycle process.
CO2: Determine software application security vulnerabilities and analyze attack consequences.
CO3: Apply secure design principles for developing attack resistant software.
CO4: Analyze insecure software, utilizing automated code review tools with static analysis and symbolic execution.
CO5: Compare tools and techniques for testing software resilience.

Suggested Books

Books Recommended

1. "SOFTWARE SECURITY: Building security In Gary McGraw", Addison – Wesley Software Security Series, 2006
2. Software Security Engineering: Julia H. Allen, Pearson Education
3. Developing Secure Software: Jason Grembi, Cengage Learning
4. Software Security : Richard Sinn, Cengage Learning

Course Code: SE 706-19	**Course Name:** Secure Software Development Lab	L:0; T:0; P:2	1 Credits

Course Objectives:
The students should be able to describe and implement various software security principles and guidelines. Laboratory experiments would enable them to mitigate security risk from external and internal sources

SYSTEM REQUIREMENTS
Open source Tools:
StarUML / UMLGraph / Topcased/Umberollo

EXPERIMENT TASK LIST

Prepare the following documents and develop the software project startup, prototype model, using software engineering methodology for a real time scenario

TASK1: Development of problem statement.
TASK 2: Identify Software quality attributes Security requirement gathering guidelines for your problem
TASK 3: Mention threats to software security and Sources of software insecurity for your chosen scenario.
TASK4: Specify the desired security properties of your problem
TASK5: Gather Requirements for your secure software and prioritize them.
TASK6: Develop use case diagrams and activity diagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams
TASK7: Perform architectural risk analysis on your problem and list attack patterns.
TASK8: Develop prototype of your problem
TASK9: Design and execute black box security test cases for your problem.
TASK10: Design and execute white box security test cases for your problem.

Course Outcomes: At the end of the course, students will be able to:

CO1: Apply Secure Software Development Life Cycle process to a real life problem.
CO2: Identify software application security vulnerabilities and analyze attack consequences.
CO3: Implement secure design principles for developing attack resistant software.
CO4: Design risk architecture model for developing secure software.
CO5: Design various test cases for secure software testing.

Reference:

1. Roger S.Pressman, Software engineering- A practitioner's Approach, McGraw-Hill International Edition, 6th edition, 2001
2 Mark Curphey, Bill Hau- Practical Software Development , OReilly
3. Michael Howard, Steve Lipner -Security Development Lifecycle, 2006, Microsoft Press