ATLAS SKILLTECH UNIVERSITY
uGDX School of Technology
Courses Offered
Core Courses, Elective Courses and Ability Enhancement Courses

Bachelor of Technology
(Computer Science, Artificial Intelligence and Machine Learning)
4 Year Full Time Program

(Academic Year 2023-24)

PREAMBLE

The University Grants Commission (UGC) has initiated several measures to bring equity, efficiency, and excellence to the Higher Education System of the country. The important measures taken to enhance academic standards and quality in higher education include innovation and improvements in curriculum, teaching-learning process, examination and evaluation systems, governance, and other matters. The UGC has formulated various regulations and guidelines from time to time to improve the higher education system and maintain minimum standards and quality across the Higher Educational Institutions (HEIs) in India. The academic reforms recommended by the UGC recently have led to overall improvement in the higher education system. However, due to a lot of diversity in the system of higher education, there are multiple approaches followed by universities towards examination, evaluation, and grading systems. While the HEIs must have the flexibility and freedom in designing the examination and evaluation methods that best fit the curriculum, syllabi, and teaching-learning methods, there is a need to devise a sensible system for awarding the grades based on the performance of students. Presently the performance of the students is reported using the conventional system of marks secured in the examinations or grades or both. The conversion from marks to letter grades and the letter grades used vary widely across the HEIs in the country. This creates difficulty for academia and employers to understand and infer the performance of the students graduating from different universities and colleges based on grades. The grading system is considered to be better than the conventional marks system and hence it has been followed in the top institutions in India and abroad. So it is desirable to introduce a uniform grading system. This will facilitate student mobility across institutions within and across countries and also enable potential employers to assess the performance of students. To bring in the desired uniformity, in the grading system and method for computing the cumulative grade point average (CGPA) based on the performance of students in the examinations, the UGC has formulated these guidelines.

CHOICE BASED CREDIT SYSTEM

The CBCS provides an opportunity for the students to choose from the prescribed courses comprising core, elective/minor, or skill-based courses. The courses can be evaluated following the grading system, which is considered to be better than the conventional marks system. Therefore, it is necessary to introduce uniform grading system in the entire higher education system in India. This will benefit the students to move across institutions within India to begin with and across countries. The uniform grading system will also enable potential employers in assessing performance of the candidates. In order to bring uniformity in the evaluation system and computation of the Cumulative Grade Point Average (CGPA) based on student's performance in examinations, the UGC has formulated the guidelines to be followed.

OUTLINE OF CHOICE BASED CREDIT SYSTEM

1. Core Course: A course, which should compulsorily be studied by a candidate as a core requirement is termed as a core course. The structure of course is defined under the following points

1.1.All the UG degree programs shall be of either six semesters or eight semesters of duration unless specified otherwise.

1.2. An academic year consists of two semesters: Odd Semester and an Even Semester.

1.3. A semester normally extends over 16 weeks (5 days a week) with 80 working days.

1.4. Every course offered may have three components: Lecture (L), Tutorial (T), and Practical (P). Tutorial session consists of participatory discussion/self-study/ desk work/ brief seminar presentations by students and other novel methods.

1.5. The credit pattern for a course (L: T:P) shall be decided by the respective
Board of Studies (BoS).

1.6. Credit means the unit by which the coursework is measured. A one-hour session of Lecture or Tutorial per week for 16 weeks amounts to 1 credit.

1.7.Two hour session of Practicals per week for 16 weeks amounts to 1 credit per semester. The total duration of a semester is 20 weeks inclusive of the semester-end examination.

1.8.A course of 3 to 6 credits will be evaluated for 100 marks. A course with less than 3 credits will be evaluated for 50 marks. For any other approved course, the evaluation method shall be decided by the respective BoS.

2. ELECTIVE COURSE:

Elective Course is a course which can be chosen from a pool of courses. It may be very specific or specialized or advanced or supportive to the discipline/ subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the student's proficiency/skill.

2.1.Discipline Specific Elective (DSE) is a course offered under the main discipline/subject of study or a Project/Dissertation.

2.2.Project/Dissertation is an elective course designed to acquire special/ advanced knowledge, such as supplement study/ support study to a project work. A student has to study such a course on his/her own with advisory support of a faculty member.

2.3.Generic Elective (GE) is an elective course chosen from an unrelated discipline/subject with an intention to seek exposure beyond discipline/subject.

3. Ability Enhancement Courses (AEC): Ability Enhancement Courses may be of two types: Ability Enhancement Compulsory Courses (AECC) and Skill Enhancement Courses (SEC).

3.1.AECC courses are mandatory courses based upon the content that leads to knowledge enhancement viz., Environmental Science, Indian Constitution and English/ Modern Indian Languages (MIL) / Communication skills.

3.2. SEC courses are aimed at providing hands-on-training, competencies, skills, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

PEO 1	Apply theoretical concepts to real-world problems with strong and efficient computer science, machine learning and artificial intelligence skills with emphasis on data driven decision making.
PEO 2	Innovate and apply critical analysis in developing solutions, integrating model-based approaches, and using advanced tools and techniques for effective problem resolution.
PEO 3	Work in multidisciplinary environments and be responsive to the changing needs of the society and evaluate the environmental, societal, ethical, economic, health and safety implications of AI and ML applications.

PROGRAM OUTCOMES (PO)

PO	Program Outcome Skill	Program Outcomes (PO)
PO1	Domain Knowledge and STEM Skills	Demonstrate domain-specific knowledge coupled with a foundation in science, technology, engineering, and mathematics (STEM), enhancing the ability to apply computing skills in various professional domains.
PO2	Experimental Design and Data Analysis Skills	Demonstrate deep understanding of data science principles, thereby designing experiments and simulations, and analyzing and interpreting datasets for informed decision-making in various applications.
PO3	Development of Practical Solutions	Master programming languages and paradigms, showcasing the ability to develop efficient and scalable software solutions, considering realistic constraints such as economic, environmental, social, ethical, health and safety, and sustainability.
PO4	Individual and Collaborative Teamwork	Function effectively as an individual, and as a member or leader in diverse and inclusive teams and in multidisciplinary, face-to-face, remote and distributed settings.
PO5	Problem Solving Skills	Develop strong problem-solving abilities, applying critical thinking, algorithmic thinking, and computational and model-based approaches to address complex challenges in diverse domains.
PO6	Ethics and Social Responsibility	Understand and apply ethical principles in computing, AI and ML, considering the societal impact of the work and making ethically informed decisions.
PO7	Effective Communication Skills	Learn to communicate complex technical concepts through powerful data visualizations, facilitating collaboration, and translating technical information for diverse audiences.
PO8	Environment and Sustainability	Understand and assess reliability, dependability and trustworthiness of AI and ML based systems and their impact in a global, economic, environmental and societal context.
PO9	Lifelong Learning	Recognize the need for, and have the ability to, engage in independent and lifelong learning and critical thinking in the

		face of rapidly involving technological landscape that is impacting all aspects of work and life across all disciplines.
PO10	Research and Development Skills	Be aware of contemporary issues facing industry and society, and engage in research and development activities, contributing to advancements in AI and ML through innovative projects and solutions.
PO11	Usage of Advanced Tools and Techniques	Apply latest AI and ML techniques, skills and tools to real-world scenarios, exhibiting competence in data engineering, developing models for data analysis, pattern recognition and decision making, and managing and deploying software solutions in multidisciplinary real-world environments.
PO12	Life Skills	Develop essential life skills, including adaptability, resilience, and a growth mindset for navigating dynamic professional environments and contributing meaningfully to society.

PROGRAM SPECIFIC OUTCOMES (PSO)

	Program Specific Outcomes (PSO)
PSO1	Demonstrate mastery in foundational skills for informed decision-making, uncertainty assessment, designing algorithms to develop robust and reliable machine learning models.
PSO2	Demonstrate ability to explore advanced topics in natural language processing, computer vision, and reinforcement learning for development of sophisticated AI addressing a wide range of tasks in diverse industries.
PSO3	Demonstrate knowledge in effectively managing and processing large datasets, deploying scalable models in cloud environments, and ensuring seamless integration of AI solutions into practical applications with ethical, societal, economic, environmental, health and safety considerations.

B. TECH (CS, AI and ML): FOUR-YEAR (8-SEMESTER)

Course Code	**Course**	**Course Type**	**Periods**			**Evaluation Scheme**				**Credits**
			L	**T**	**P**	**MT**	**ET**	**CIA**	**Total**	
Semester 1										
BTECH MAT 101	Basics of Calculus 1	Core	**3**			**20**	**40**	**40**		3
BTECH MAT 103	Foundations of Statistics and Probability for AI and ML	Core	**1**		**1**	**20**	**40**	**40**		2
BTECH CAP 101	Foundational Maths for Computer Science	Core	**3**			**20**	**40**	**40**		3
BTECH CAP 103	Databases and SQL	Core	**2**		**1**	**20**	**40**	**40**		3
BTECH LA 101	Expository Writing	AEC	**2**			**20**	**40**	**40**		2
BTECH ASC 101	Chemistry	VAC	**2**			**20**	**40**	**40**		2
BTECH CAP 105	Introduction to Computer Science and Programming 1	Core	**1**		**2**	**20**	**40**	**40**		3
AE	ATLAS Electives	Multidisciplinary	**2**							2

BTECH CPT 104	Games and Puzzles	SEC			**1**					1
	TOTAL									**21**
Semester 2										
BTECH CAP 102	Computer Assembly and Architecture	Core	**2**		**1**	**20**	**40**	**40**		3
BTECH MAT 102	Basics of Calculus 2	Core	**3**			**20**	**40**	**40**		3
BTECH DSC 102	Statistical Modeling 1	Core	**1**		**1**	**20**	**40**	**40**		2
BTECH MAT 104	Basics of Linear Algebra	Core	**4**			**20**	**40**	**40**		4
BTECH CAP 104	Introduction to Computer Science and Programming 2	Core	**1**		**2**	**20**	**40**	**40**		3
BTECH LA 102	Model Thinking	AEC	**2**			**20**	**40**	**40**		2
AE	ATLAS Electives		**2**							2
BTECH CPT 106	Data Analyst Project	SEC			**1**					1
	TOTAL									**20**
Semester 3										
	Computer Networks	Core	**2**		**2**	**20**	**40**	**40**		3

	Data Visualization and Business Intelligence	Core	**1**		**2**	**20**	**40**	**40**		3
	Unstructured Data, NoSQL and Storage Techniques	Core	**2**		**1**	**20**	**40**	**40**		3
	Data Structures	Core	**2**		**1**	**20**	**40**	**40**		3
	Statistical Modelling -2	Core	**1**		**1**	**20**	**40**	**40**		2
	Machine Learning: Methods and Algorithms - 1	Core	**1**		**1**	**20**	**40**	**40**		2
	Physics	VAC	**2**		**1**	**20**	**40**	**40**		3
	ATLAS Electives	Multidisciplinary	**2**							2
	Biology	VAC	**2**			**20**	**40**	**40**		2
	Machine Learning: Astrophysics, Particles, Drug Design.	SEC			**2**					1
	TOTAL									**24**
Semester 4										
	Large-Scale Data Storage	Core	**2**		**1**	**20**	**40**	**40**		3
	Data Engineering Operations	Core	**2**		**1**	**20**	**40**	**40**		3

	Introduction to Algorithms	Core	**2**		**1**	**20**	**40**	**40**		3
	Writing Business Plans, Technical Papers and Blogs	AEC	**2**			**20**	**40**	**40**		2
	Machine Learning: Methods and Algorithms - 2	Core	**2**		**1**	**20**	**40**	**40**		2
	Software Development Lifecycle, DevOps and MLOps	Core	**3**		**1**	**20**	**40**	**40**		4
	ATLAS Electives	Multi	**2**							2
	An Enterprise Grade AI Application: Recommendation Engines for OTT Platforms	SEC			**1**					1
	TOTAL									**20**
Semester 5										
	Reinforcement Learning	Core	**2**		**1**	**20**	**40**	**40**		3
	Computer Vision	Core	**2**		**1**	**20**	**40**	**40**		3
	Deep Learning and NLP	Core	**2**		**1**	**20**	**40**	**40**		3

	Digital Technologies, Smart Products and Services	Core	**3**		**1**	**20**	**40**	**40**		4
	Operations Research	Core	**2**		**1**	**20**	**40**	**40**		3
	Advanced Algorithms	Core	**3**			**20**	**40**	**40**		3
	Cloud Application Development	Core	**2**		**1**	**20**	**40**	**40**		3
	ATLAS Electives	Multi	**2**							2
	Enterprise Grade Connected Device Application: Self-Driving Cars	SEC			**1**					1
	TOTAL									**25**
Semester 6										
	Internship	Internship								20
	TOTAL									
Semester 7										
	UI/UX Development	**Core**	**2**		**1**	**20**	**40**	**40**		**3**
	Secure Coding	**Core**	**2**		**1**	**20**	**40**	**40**		**3**

	Networks and Network Security	**Core**	**2**		**1**	**20**	**40**	**40**		**3**
	Entrepreneurship Development	**AEC**	**2**			**20**	**40**	**40**		**2**
	Recent Advances and Challenges in AI	**Core**	**3**			**20**	**40**	**40**		**3**
	Ethical Issues in Adopting AI	**SEC**	**2**			**20**	**40**	**40**		**2**
	Final Year Design Project 1	**Project**								**3**
	Build Your Own Encryption Software	**SEC**	**1**							**1**
	TOTAL									**20**
Semester 8										
	AI in Engineering	**Core**	**3**		**1**	**20**	**40**	**40**		**4**
	Cybersecurity and Blockchain	**Core**	**2**		**1**	**20**	**40**	**40**		**3**
	Intellectual Property and Innovation in AI	**AEC**	**3**			**20**	**40**	**40**		**3**
	Ethical Hacking	**Core**	**2**		**1**	**20**	**40**	**40**		**3**

	Final Year Design Project 2	**Project**								**8**
	TOTAL									**21**

B. TECH FIRST YEAR
SEMESTER 1

L	T	P
3		

COURSE NAME: Basics of Calculus 1 **COURSE CONTACT HOURS: 3**
COURSE CREDIT HOURS: 3 **COURSE CODE: BTECH MAT 101**

COURSE DESCRIPTION:

This course is intended to develop practical skills in differential calculus. Calculus plays an important role in science, engineering, computer science, data engineering, artificial intelligence, economics, finance, and many other fields. Topics covered in this course include review of functions and trigonometric functions, limits, rates of change, differentiation, the mean value theorem, and applications of derivatives.

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

This course is the first course in a four-course calculus sequence. Students who successfully complete this course may continue into "Calculus - 2", which is taught in the second semester.
The objectives of this course are the following: to enable students to become a competent user of differential calculus, to enable them to develop the ability to write well-organized, coherent solutions to problems, to enable them to understand the concept of derivative as representing rate of change and slope, and to position students so that they can excel in subjects with calculus components.

COURSE OUTCOMES (CO'S)

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

1. Evaluate limits and determine continuity properties of functions (CO1)
2. Calculate derivatives using a variety of methods (CO2)
3. Analyze the connection between derivatives and tangent lines, and find the equation of the tangent line to a graph at a given point (CO3)
4. Interpret derivatives across several fields of physical sciences and engineering (CO4)
5. Solve maximum and minimum problems using differentiation (CO5)
6. Apply methods of calculus to curve sketching (CO6)

LEARNING RESOURCES

Required Resources

- George B. Thomas, Jr. (with revisions by Joel Hass, Christopher Heil and Maurice D. Weir), “Thomas’ Calculus”, 14th Edition, Pearson, 2018.

Recommended Resources:

- N/A

COURSE CONTENT

Sr.No	Modules/ Units
1	Functions
2	Limits and continuity
	Rates of change and tangent lines to curves Limit of a function and limit laws The precise definition of a limit One-sided limits Continuity Limits involving infinity
3	Derivatives
	Tangent lines and the derivative at a point The derivative of a function Differentiation rules The derivative as a rate of change Derivatives of trigonometric functions The chain rule Implicit differentiation Related rates Linearization and differentials
4	Applications of derivatives
5	Extreme values of functions on closed intervals The mean value theorem Monotonic functions and the first derivative test Concavity and curve sketching Applied optimization Newton’s method Antiderivatives

L	T	P
1		1

COURSE NAME: FOUNDATIONS OF STATISTICS CONTACT HOURS: 3

AND PROBABILITY FOR AI AND ML COURSE

COURSE CREDIT HOURS: 2 COURSE CODE: BTECH MAT 103

COURSE DESCRIPTION:

This is a foundational course covering important topics that form the basis of the entire field of Machine Learning and Artificial Intelligence. Through this course, students will develop a deep understanding of key statistical concepts such as central tendencies, measures of variability, probability distributions, Central Limit Theorem, confidence intervals and hypothesis testing, enabling them to effectively analyze and interpret data, and draw inferences from sample data. By exploring the principles of probability theory, learners will gain the skills necessary to model uncertainty, make informed decisions, and gain skills to design algorithms that drive AI and machine learning systems. This course forms the bedrock upon which students can confidently build their knowledge in artificial intelligence and machine learning, empowering them to create sophisticated models, optimize algorithms, and extract meaningful insights from complex datasets.

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

Students who successfully complete this course are prepared to build models for predicting outcomes based on input data, which they will learn in subsequent semesters.

The objectives of this course are:

- To perform basic analysis and summarize datasets for understanding patterns in the data
- To understand Probability Theory for Data Science/AI/ML
- To understand the distribution of data and important distributions
- To understand Central Limit Theorem, Confidence Levels, Confidence intervals and Hypothesis testing for making inferences from sampled data

COURSE OUTCOMES (CO'S)

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

- Compute various measures of descriptive data analysis including variability and spread (CO01).
- Solve probability problems based on the type of probability (CO02).
- Understand probability distributions, their applications and compute probabilities and random variable's values using the distributions (CO03).
- Compute confidence intervals at different levels of statistical significance (CO4).
- State the null and alternate hypothesis, find a suitable statistic for a given problem and make inferences for some of the challenging real-world problems (CO05).
- Perform various types of statistical tests on data for making inferences (CO06).

LEARNING RESOURCES

Required Resources

- None

Recommended Resources:

- Applied Business Statistics by Ken Black, 7th edition, Wiley, 2012. ISBN-13: 978-8126537075; ISBN-10: 9788126537075
- Statistics For Business: Decision Making and Analysis by Robert Stine and Dean Foster, 3rd edition, Pearson, 2020. ISBN-13: 978-9353940645; ISBN-10: 9353940648
- Statistics for Absolute Beginners, O Theobald, 2nd edition, Scatterplot Press, 2020
- http://stattrek.com
- http://www.khanacademy.org
- http://vassarstats.net/textbook
- http://onlinestatbook.com/rvls.html
- https://www.itl.nist.gov/div898/handbook/index.htm
- https://blog.minitab.com/blog

COURSE CONTENT

Sr.No	Modules/ Units
1	**Motivation and Statistics Basics**
	Why study Statistics? Population and Sample Parameter and Statistic Descriptive and Inferential Statistics Variables and Data: Numeric, Categorical (Nominal, Ordinal) Discrete and Continuous

2	**Describing Data Through Statistics**
	Central Tendencies Measures of Variability Box Plot and Outlier Detection
3	**Probability Basics**
	Probability vs Statistics Frequentist Approach to Probability Computation Probability Rules, Mutually Exclusive Events and Independent Events Joint, Union, Marginal and Conditional Probabilities Bayes Theorem
4	**Probability Distributions**
	Conceptual Understanding Binomial Distribution Normal Distribution z-Distribution (Standard Normal Distribution)
5	**Inferential Statistics**
	Sampling Distribution of Means Central Limit Theorem Confidence Intervals Hypothesis Testing
6	**Statistical Tests**
	t-Tests Chi-Square Tests F-Test

L	T	P
3		

COURSE NAME: Foundational Maths for Computer Science COURSE CONTACT HOURS:3

COURSE CREDIT HOURS: 3 **COURSE CODE : BTECH CAP 101**

COURSE DESCRIPTION:

This course offers an introduction to discrete mathematics oriented toward computer science and engineering. Topics covered include the following: logic and proofs, sets and functions, number theory, proof by induction, counting techniques, recurrences, relations, and graphs.

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

After completing this course, students will be able to explain and apply basic methods of discrete (noncontinuous) mathematics in computer science. These methods will be used in subsequent courses in data structures, design and analysis of algorithms, computation, software engineering and computer systems.

COURSE OUTCOMES (CO'S)

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

1. Communicate logically rigorously, prove mathematical statements with logical rigor, and identify fallacious reasoning (CO1)
2. Use logical notation to define and reason about fundamental mathematical concepts such as sets and functions (CO2)
3. Prove elementary properties of modular arithmetic and explain their applications in computer science, for example, in cryptography and hashing algorithms (CO3)
4. Synthesize induction hypotheses and simple induction proofs (CO4)
5. Calculate numbers of possible outcomes of elementary combinatorial processes such as permutations and combinations (CO5)
6. Derive closed-form and asymptotic expressions from series and recurrences for growth rates of processes (CO6)
7. Prove properties of relations and graphs and model real-world problems using relations and graphs (CO7)

LEARNING RESOURCES

Required Resources

- Kenneth Rosen, "Discrete Mathematics and its Applications," Seventh Edition, McGraw Hill, 2012.

Recommended Resources:

- "Mathematics for Computer Science", MIT Open Courseware, 6.042J, Fall 2010. Course notes at https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/pages/readings/

COURSE CONTENT

Sr.No	**Modules/ Units**
1	**Foundations: logic and proofs**
	Proposition logic and equivalences Predicates and quantifiers Rules of inference Introduction to proofs
2	**Sets, functions, number theory, induction**
	Sets and set operations Functions Divisibility and modular arithmetic Primes and greatest common divisors Induction and strong induction
3	**Counting**
	Basics of counting Pigeonhole principle Permutations and combinations Applications of recurrence relations

	Solving linear recurrence relations Generating functions
4	**Relations**
	Relations and their properties Representing relations Equivalence relations Partial orderings
5	**Graphs**
	Graphs and graph models Graph terminology and special types of graphs Representing graphs and graph isomorphism Connectivity Euler and Hamilton paths Planar graphs Graph coloring

L	T	P
2		**1**

COURSE NAME: DATABASES AND SQL **COURSE CONTACT HOURS: 4**

COURSE CREDIT HOURS: 3 **COURSE CODE: BTECH CAP 103**

COURSE DESCRIPTION:

This course is intended to develop practical skills in designing relational database management system. It helps to understand in detail about the architecture of database management system, designing a database, fine tuning the database using normalization concepts. Also, this course describes about the storage concepts, transaction management, concurrency control concepts and recovery options for database management system.

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

The objectives of the course are as follows:

- To enable the students to familiarize the concepts of database architecture and design
- To design a database using ER diagram and implement using SQL
- To learn about the basics of dependency and normalization
- To understand the concepts of transactional management
- To understand about the storage and file structures
- To understand about the issues with concurrency control and failure recovery

COURSE OUTCOMES (CO'S)

After completion of the course, students should be able to:

1. Understand in-depth the concepts of database architecture and design (CO1)
2. Design a database using ER diagram and normalize it (CO2)
3. Convert the ER model to a relational database and apply SQL queries (CO3)
4. Obtain an in-depth understanding of storage and file structures (CO4)
5. Identify various issues with respect to transaction management (CO5)
6. Develop competency in handling issues related to concurrency control and failure recovery (CO6)

LEARNING RESOURCES

Reference Book

1. A Silber Schatz, H Korth and S Sudarshan, Database System Concepts, 6th Ed., McGraw-Hill, 2010.
2. R Elmasri, S Navathe, Fundamentals of Database Systems, 6th edition, Addison-Wesley, 2010.

Recommended Resources:

1. H Garcia-Molina, JD Ullman and Widom, Database Systems: The Complete Book,2nd Ed., Prentice-Hall, 2008.
2. R Ramakrishnan, J Gehrke, Database Management Systems, 3rd Ed., McGraw-Hill, 2002.

Modules	
1. Introduction	
1.1	Database Applications, Purpose, Views of data
1.2	Database Languages, Database Design
1.3	Architecture - Users and Administrators, Data Mining, Information Retrieval
1.4	Relational Databases: Relational model, Database Schema
1.5	Keys, Relational Query Languages
2. Database Design	
2.1	E-R model - E-R diagram
2.2	Reduction to Relational Schema, E-R design issues,
2.3	Database Integrity, Specifying Integrity constraints in SQL
2.4	Unique columns, Foreign Key, Triggers.

2.5	Relational Database Design: Features of good design,
2.6	Functional Dependency Theory, Decomposition using Functional Dependency
2.7	Normal forms, Algorithms for Decomposition
2.8	Multi-Valued Dependency, and 4th Normal Form
3. Storage, Indexing and Transaction Management	
3.1	Storage and File structure: Overview of secondary storage, RAID, and flash storage
3.2	Indexing: Basic Concepts - Ordered Indices
3.3	B+ trees, Transaction Management: Concept and purpose
3.4	ACID properties and their necessity, transactions in SQL
3.5	Problems with full isolation and levels of isolation
4. Concurrency Control and Recovery	
4.1	Lock-based Protocols, 2-phase locking
4.2	Deadlock handling, Multiple granularity
4.3	Timestamp Based Protocols, Index Locking
4.4	Recovery: Failures and their Classification
4.5	Recovery and Atomicity, Recovery algorithms, Undo-Redo with write ahead logging

4.6	No Undo No Redo and other combinations, buffer management, ARIES recovery
5. SQL	
5.1	SQL: Data definition, basic SQL query structure - DML
5.2	Set Operations, Nested Subqueries
5.3	Aggregation, Null values
5.4	Database Modification, Join Expressions
5.5	Views, Date and Time functions, Stored procedures, and Triggers

L	T	P
2		

COURSE NAME: EXPOSITORY WRITING **COURSE CONTACT HOURS: 2**
COURSE CREDIT HOURS: 2 **COURSE CODE: BTECH LA 101**

COURSE DESCRIPTION:

This class introduces the comprehensive skills of expository writing and critical reading. Students will learn to analyze various types of reading material and will strengthen their writing skills. These skills will support their academic work and prepare them to meet the reading and writing demands of professional activities.

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

The objectives of this course are:

- Think critically about a text to evaluate its persuasiveness.
- Identify and summarize main ideas.
- Create an argument based on evidence.
- Communicate with an academic or professional audience through clear and coherent writing.
- Cite sources according to APA and MLA format.

COURSE OUTCOMES (COS)

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

- Recognize and differentiate between various forms of writing, understanding specifically what constitutes expository writing. (CO01)
- Structure their writing logically, making use of techniques such as cause and effect, comparison and contrast, and problem and solution to arrange their ideas. (CO02)
- Clear, concise, and arguable thesis statements that guide their expository essays. (CO03)
- Support their claims with relevant, credible, and varied evidence, understanding how to incorporate sources, facts, statistics, and expert opinions. (CO04)
- Understand the importance of revising and editing, and learning techniques to improve the clarity, coherence, and overall quality of their writing. (CO05)
- Learn how to conduct research effectively, discerning between credible and non-credible

sources, and integrating their findings into their essays with proper citation. (CO06)

- Understand the importance of avoiding plagiarism and will know how to cite sources correctly. (CO07)
- Will engage in peer review processes, both giving and receiving constructive feedback, to hone their writing and analytical skills. (CO08)

LEARNING RESOURCES

Required Resources

- Will be provided by the faculty

Recommended Resources:

- None

COURSE CONTENT

Sr.No	Modules/ Units
1	**Essay Writing and Reviewing**
	● Fundamental Writing Rules ● Readerly Habits ● APA and MLA Citations ● Annotation and Close Reading ● Structure of an Essay ● Textual evidence and developing a working thesis ● Structuring a paragraph and incorporating textual evidence ● Writing Process ● Revision and Peer Review ● Essay Grading Rubric
2	**Research and Evidence**
	● Discussing outside contexts and incorporating research into argument. ● Logical connectives and transitions in essays ● Reverse outline and revision. ● Developing paragraphs and integrating sources

3	**Critical Review and Submission**
	● Concept of critical fabulation ● Hartman’s excerpt and research ● Editing and conclusions ● Presenting your ideas.

L	T	P
2		

COURSE NAME: CHEMISTRY
COURSE CREDIT HOURS: 2

COURSE CONTACT HOURS: 2
COURSE CODE: BTECH ASC 101

COURSE DESCRIPTION:

This course forms a part of the curriculum for Basic Sciences. It provides a fundamental understanding of the building blocks of nature and the interactions that binds them. Since Data Science is an inter-disciplinary subject, students pursuing career in Data Science are expected to have a basic understanding of the Natural Sciences. The pedagogy is designed to benefit students with very basic knowledge in Physical Sciences at 10+2 level. The course would also walk the students through the state-of-the-art progress of Machine Learning and Data Science in the field of Industrial and Fundamental Chemistry. With freely distributed online softwares, this course would enable the design of macro-molecules from their respective chemical blocks.

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

COURSE OBJECTIVES

- Comprehension of the Fundamental Building Blocks of Nature and their Interactions
- Acquaintance with the state-of-the-art implementation of the acquired knowledge.
- Hands-On knowledge of Molecular Design and the estimation of Physical Properties of complex molecules using free softwares.

COURSE OUTCOMES

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

1) Have a generic understanding of the fundamental properties of matter
2) Have a general overview of different chemical substrates and their intrinsic traits
3) Working knowledge of the implementation of colloids, acids, alkalies in the industries
4) Estimation of molecular structures from their respective basic constituents using online softwares

LEARNING RESOURCES

Required Resources:

None. The Class Lectures, board works and the demonstrations shown in the class would be adequate and substantial.

Recommended Resources:

“Chemistry: A Molecular Approach” by Nivaldo J. Tro, Fifth Edition, Pearson.

COURSE CONTENT

Sr.No	Modules/ Units
1	Atomic Structure: Dalton’s Theory, Thomson’s discovery, Rutherford’s Model, Bohr’s Postulate,

	Quantum Numbers of an electron, Aufbau's Principle, Pauli's Exclusion Principle, Hund's Rule.
2	Chemical Bonding: Ionic Bonds, Covalent Bonds, Coordinate Bonds, Hybridization, A brief overview of Molecular Orbital Theory, Hydrogen bonds, Van-Der-Waal's Force and molecularization of Inert gases.
3	Periodic Table: Trends and Recurrences of Physical and Chemical Properties, Variations of these properties along the row and the column in a Periodic Table, Transition Elements, Actinides and Lanthanides, Acidic and Basic Properties, Metallic and Non-metallic properties, Inert gases.
4	Chemical Equilibrium and Electrochemistry: Equilibrium Constant, Direction of a Reaction, Le Chatelier's Principle, Batteries and Electromotive Force, Electrolysis.
5	Solutions and Colloids: Solution Stoichiometry, Solubility, Types of Solution, Types of Solvent, Aqueous reactions, Redox reactions.

L	T	P

2		1

COURSE NAME: INTRODUCTION TO COMPUTER SCIENCE AND PROGRAMMING 1
COURSE CONTACT HOURS: 4
COURSE CREDIT HOURS: 3
COURSE CODE: BTECH CAP 105

COURSE DESCRIPTION:

This course is an introduction to computer programming. No prior programming experience is assumed. Students will learn to design, write, and debug computer programs in Python. Topics covered include variables and data types, tuples, lists, dictionaries, input and output methods, control structures, functions and testing and debugging. This course will introduce students to computational approaches to problem solving using Python. Modern software engineering principles, especially modularity, will also be emphasized. By the end of the course, students will have learnt to write their own functions and programs in Python.

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

Computational thinking is the process of analyzing a problem, and then designing and expressing its solution in suh a way that a computer can effectively solve the problem. Computational thinking includes a number of characteristics, such as breaking a problem into small and repetitive ordered steps, logically ordering and analyzing data, and creating solutions that can be effectively implemented as programs running on a computer.

The aim of this course is hence to take students with no prior experience of thinking in a computational manner to a point where they can derive simple algorithms and code the programs to solve some basic problems in their chosen domains. Student will also learn about Python variables and expressions, looping mechanisms, functions, lists, and dictionaries. While Python syntax will be taught, the objective is really to teach computational thinking and problem solving rather than the syntax of any particular programming language.

COURSE OUTCOMES (CO'S)

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

1. Apply basic concepts of computational thinking, including sequential logic, abstractions, conceptualization and problem-solving (CO1)
2. Develop computer solutions to problems of low-to-moderate complexity (CO2)
3. Implement solutions using suitable data types and constructs in Python (CO3)
4. Model real-life problems as computational problems (CO4)

LEARNING RESOURCES

Required Resources

- John Guttag, “Introduction to Computation and Programming Using Python: With Application to Understanding Data,” Second Edition, PHI, 2016.

Recommended Resources:

- Donald Knuth, “The art of computer programming”, Volumes 1, Fundamental Algorithms, Addison-Wesley, 1968.
- Mark Lutz, “Learning Python”, 5th Edition, O’Reilley, 2013.

COURSE CONTENT

Sr.No	Modules/ Units
1	**Introduction to Machines and Languages**
2	**Introduction to Python**
	● Objects, expressions ● Numerical types, variables and assignment ● Python IDE’s ● Branching programs ● Strings and input
3	**Some simple numerical programs**
	● Exhaustive enumeration ● For loops ● Approximate solutions ● Bisection search ● Floats
4	**Functions, scoping, and abstraction**
	● Functions and Scoping ● Specifications ● Recursion ● Global Variables ● Modules ● Files

5	**Structured types, mutability, and higher-order functions**
	● Tuples ● Ranges ● Lists and mutability ● Functions as objects ● Strings, tuples, ranges, and lists ● Dictionaries

B. TECH 1ST YEAR
SEMESTER 2

L	T	P
2		1

COURSE NAME: COMPUTER ASSEMBLY AND ARCHITECTURE
COURSE CONTACT HOURS: 4
COURSE CREDIT HOURS: 3 **COURSE CODE: BTECH CAP 102**

COURSE DESCRIPTION:

This Computer Organization and Architecture course offers an in-depth study of computer architecture and system management, tailored for computer science students. It focuses on understanding computer system components such as CPUs, memory, and I/O devices, and applying performance metrics for system evaluation. The course delves into various computer architectures, examining their impact on performance. Key topics include process management, covering scheduling, synchronization, deadlock resolution, and memory management techniques. Through a blend of theory and practical application, including lectures and lab sessions, students gain essential skills for optimizing computer system efficiency, preparing them for advanced studies or careers in technology.

Pre-requisites**:** Basic Knowledge of Digital Electronics and Microprocessors

COURSE OBJECTIVES AND COURSE OUTCOMES

COURSE OBJECTIVES

Students who successfully complete this course are prepared to build statistical models for some of the most common types of real-world Data Science problems – Prediction, Classification and Forecasting.

The objectives of this course are:

1. Understand Systems in Software Development: Grasp the influence of computer organization and operating systems on software development.
2. Analyze Computer Architecture Impact: Learn how computer architecture affects software performance and design. Explore OS Principles for Software Development: Understand operating system concepts and their impact on software functionality.
3. Link Software Performance to System Features: Develop skills to correlate software performance with system-level features like architecture and OS design.

4. Apply Theory Practically: Enhance problem-solving skills in software development, considering system constraints.

COURSE OUTCOMES

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

- **CO1: Develop a comprehensive understanding of the basic components of computer system**
- **CO2:** A**pply** the knowledge of performance metrics to find the performance of systems..
- **CO3: Examine** different computer architectures and hardware
- **CO4**: **Analyse** and compare of process management concepts including scheduling, synchronization, deadlocks.
- **CO5: Analyse** and compare various memory management techniques

LEARNING RESOURCES

Required Text Reading

- None

Books

- Stallings William, Computer Organization & Architecture, Pearson Education, 10th Ed. 2013
- A Silberschatz, Abraham and others, Operating Systems Concepts, Wiley Student Edition, 9th Ed.Norman Matloff.
- Patterson, David A & J L Hennenssy, Computer Organization and Design – The Hardware/Software Interface, Elsevier, Revised 4th Ed.
- Randal E. Bryant, David R. O'Hallaron, Computer Systems – A Programmer's Perspective, Pearson, 2nd Ed, 2016.
- Kai Hwang and Briggs, Computer Architecture and Parallel Processing, Tata McGrawHill Edition
- Stallings, Operating Systems: Internals and Design Principles, International Edition, Pearson Education, 2013 (Pearson Online)

Modules	
Introduction to Computer Systems	
1.1	Basic Operational concepts, Bus structures.
1.2	Basic uniprocessor and Multi-processor Architecture
1.3	Instruction Cycle State Diagram
1.4	Embedded Systems
1.5	Arm Architecture
1.6	Cloud Computing
1.7	Microprocessors versus Microcontrollers
1.8	Performance Assessment- MIPS Rate
1.9	Amdahl’s Law
Computer Architecture and Organization	
2.1	Von-Neumann Architecture vs Harvard Architecture
2.2	RISC Vs CISC
2.3	Machine Instruction Characteristics
2.4	Types of Operands, Operations and Addressing Modes

2.5	Instruction Formats
2.6	**Register Transfer Language and Micro Operations-Introduction**
2.7	RTL- Registers, Register transfers, Bus and memory transfers.
2.8	Micro operations: Arithmetic, Logic, and Shift micro-operations, Arithmetic logic shift unit.
2.9	Computer Registers, Computer instructions, Instruction cycle.
2.10	Instruction codes, Timing and Control, Types of Instructions: Memory Reference Instructions, Input – Output and Interrupt
Computer Organization and Design : Memory Organization, CPU and Cache	
3.1	Internal Memory - o Semiconductor Main Memory (SRAM and DRAM) o DDR – DRAM
3.2	External Memory – o Magnetic Disk, RAID o SSD - Solid State Memory Technologies (Types)
3.3	Cache Memory Organization o Locality (Class Notes) o Locality of Reference to Program Data o Locality of instruction fetches Memory Hierarchy Cache Memories o Generic Cache Memory Organization o Direct Mapped Cache
3.4	Cache Memory Organization o Fully Associative Cache o Set Associative Caches

	o Issues with Writes o Performance Impact of Cache Parameters
Process Management	
4.1	Concept of Process
4.2	Process State Diagram
4.3	Operations on Processes : Process creation and termination examples
4.4	Process vs. Threads
4.5	Process Scheduling criteria
4.6	Process Scheduling Algorithms -FCFS, SJF
4.7	**Introduction to Process Coordination**
	Semaphores
	Deadlock
	Memory and Input –Output Management
5.1	Memory-Management Strategies
5.2	Paging
5.3	Segmentation
5.4	Virtual-Memory
5.5	Demand Paging

5.6	Page Replacement Algorithms: FIFO, Optimal, LRU, and LFU
5.7	Input –Output Organization: Peripheral devices, Input-output subsystems, I/O
5.8	device interface, interrupts and exceptions. I/O device interfaces – SCII, USB

LAB CONTENT- LIST OF EXPERIMENTS

1.Introduction to Computer Hardware

- Identifying Computer Components: Familiarization with the physical components of a computer including CPU, memory, I/O devices.
  (Requirements: CPU Box-2, Connectors-BNC, RJ 45, RJ11, CAT 16 cable, Crimping tool, Nonworking Hard disks-2, Peripheral devices-Monitor, Keyboard, Mouse, Printer cable(USB and D Type)
- Building a Simple Computer: Assembling a basic computer system from components to understand the interconnections and functions.

2. Processor and Microarchitecture

- Processor Dissection: Disassembling a CPU to identify its parts and discussing its design and functionality.
- Memory Systems

3. Types of Memory

- Input/Output Systems
- I/O Interface and Buses: Exploring various I/O interfaces and bus systems, and understanding their protocols and data transfer methods.
- Peripheral Connectivity: Connecting and configuring different types of peripheral devices.

4. Operating System Fundamentals

- Linux Installation and Configuration: Installing a Linux distribution and configuring the system settings.
- Shell Scripting Basics: Writing basic shell scripts to automate tasks and understand command-line operations.

5. System Administration

- User and Group Management: Creating and managing users and groups, understanding permissions and ownership.
- Network Configuration: Setting up and managing network configurations, exploring different network tools and commands.

6. Process and Memory Management

- Process Monitoring and Management: Using tools like ps, top, htop to monitor and manage system processes.
- Memory Allocation: Exploring memory usage with commands like free, understanding virtual memory, and swap space.

7. File Systems and Storage

- File System Navigation and Manipulation: Working with the Linux file system, understanding directories, files, and file types.
- Disk Partitioning and Mounting: Creating, partitioning, and mounting disks using command-line tools.

8. Device and Module Management

- Kernel Module and Driver Installation: Loading and unloading kernel modules, understanding the role of drivers.
- Device Management: Exploring /dev directory, understanding major and minor numbers, working with block and character devices.

9. Security and Permissions

- Implementing Firewalls: Configuring and managing firewalls using iptables or ufw.
- Access Control Lists (ACLs) and Security: Setting up and managing ACLs for files and directories, understanding basic security protocols.

10. Performance Tuning and Monitoring

- System Monitoring Tools: Using tools like vmstat, iostat, netstat to monitor system performance.
- Performance Tuning: Adjusting system parameters for optimized performance, understanding nice values, and CPU affinity.

L	T	P
3		

COURSE NAME: CALCULUS 2 **COURSE CONTACT HOURS: 3**

COURSE CREDIT HOURS: 3 **COURSE CODE: BTECH MAT 102**

COURSE DESCRIPTION:

This course is a continuation of the introductory calculus course. Topics covered include integrals, applications of definite integrals, transcendental functions, and techniques of integration.

COURSE OBJECTIVES AND COURSE OUTCOMES

COURSE OBJECTIVES

Students who successfully complete this course are prepared to build statistical models for some of the most common types of real-world Data Science problems – Prediction, Classification and Forecasting.

The objectives of this course are:

1. Basics of Integration: Understand area estimation with finite sums. Explore sigma notation and limits. Grasp definite integral fundamentals.
2. Advanced Integration: Apply substitution methods. Utilize definite integral substitutions.
3. Applications of Integrals: Calculate volumes and solve related problems. Determine arc length, surface areas, and solve work-related problems.
4. Transcendental Functions: Study inverse functions and derivatives. Analyze logarithmic, exponential functions and solve such problems
5. Integration Techniques: Use basic integration formulas effectively. Apply integration by parts, trigonometric techniques. Perform partial fractions, numerical integration, and handle improper integrals.

COURSE OUTCOMES

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

CO1: Proficiency in Integration: Demonstrate skill in calculating areas. Apply definite integrals to solve practical problems.
CO2: Mastery in Integration Techniques: Successfully employ substitution methods. Apply definite integral substitutions for real-world scenarios.
CO3: Practical Applications: Solve problems related to volumes, arc length, and work. Apply definite integrals in practical scenarios.
CO4: Transcendental Function Proficiency: Analyze and differentiate inverse functions. Apply logarithmic, exponential functions to real-world problems.
CO5: Integration Skills: Effectively use basic integration formulas. Apply integration techniques to solve diverse problems.
CO6: Numerical Integrals skills: Accurately evaluate numerical integrals, improper integrals, and probability calculations, enhancing their problem-solving skills in various mathematical contexts.

LEARNING RESOURCES

Required Text Reading

- “Thomas’ Calculus”, George B. Thomas, Jr., 14th Edition, Pearson, 2018
- “Calculus”, Gilbert Strang, Wellesley-Cambridge Press, MIT Open Courseware

Modules	
Integrals	
1.1	Area and Estimating with Finite Sums
1.2	Sigma Notation and Limits of Finite Sums
1.3	The Definite Integral
1.4	The Fundamental Theorem of Calculus
1.5	Indefinite Integrals and the Substitution Method
1.6	Definite Integral Substitutions and the Area Between Curves
Applications of Definite Integrals	
2.1	Volumes Using Cross-Sections
2.2	Volumes Using Cylindrical Shells
2.3	Arc Length
2.4	Areas of Surfaces of Revolution
2.5	Work and Fluid Forces
2.6	Moments and Centers of Mass

Transcendental functions	
3.1	Inverse Functions and Their Derivatives
3.2	Natural Logarithms
3.3	Exponential Functions
3.4	Exponential Change and Separable Differential Equations
3.5	Indeterminate Forms and L’Hôpital’s Rule
3.6	Inverse Trigonometric Functions
3.7	Hyperbolic Functions
3.8	Relative Rates of Growth
Techniques of Integration	
4.1	Using Basic Integration Formulas
4.2	Integration by Parts
4.3	Trigonometric Integrals
4.4	Trigonometric Substitutions
4.5	Integration of Rational Functions by Partial Fractions
4.6	Integral Tables and Computer Algebra Systems
4.7	Numerical Integration
4.8	Improper Integrals

4.9	Probability

L	T	P
1		1

COURSE NAME: STATISTICAL MODELING -1 **COURSE CONTACT HOURS: 3**
COURSE CREDIT HOURS: 2 **COURSE CODE: BTECH DSC 102**

COURSE DESCRIPTION:

This course immerses students in fundamental predictive modeling techniques with a strong emphasis on practical applications. The course fosters a deep understanding of statistical modeling and the nuances of various algorithms, providing a robust foundation for students to navigate the dynamic landscape of AI and ML. Students gain practical insights into solving real-world problems dealing with predictions and classifications, enhancing their readiness for the evolving demands of data-driven decision-making in professional settings. The curriculum integrates hands-on experiences, enabling students to implement these algorithms using Python and apply their knowledge to real-world problems. By the course's conclusion, students emerge well-equipped to analyze complex datasets, make informed decisions, and address business challenges using statistical models.

PRE-REQUISITES:

Foundations of Statistics and Probability for AI and ML
Year and Term: First Year and Second Term

COURSE OBJECTIVES AND COURSE OUTCOMES

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

CO1: Understand and explain the concepts of prediction using simple and multiple linear regression and analyze model outputs.
CO2: Build simple and multiple linear regression models using Python, analyze model outputs, transform data appropriately, identify significant predictors and take decisions that have business impact in various domains.
CO3: Understand and explain the concepts of classification using (binary) logistic regression and analyze model outputs.
CO4: Build (binary) logistic regression models using Python, analyze model outputs, transform data appropriately, identify significant predictors and take decisions that have business impact in various domains.
CO5: Understand and explain concepts of Naïve Bayes classifier and analyze model outputs.
CO6: Build Naïve Bayes classifier using Python, analyze model outputs and take decisions that have business impact in various domains.

Required Text Reading

- None

BOOKS

- Norman Matloff. *Statistical Regression and Classification: From Linear Models to Machine Learning*, First Edition, Chapman and Hall/ CRC Press. 2017
- https://open.umn.edu/opentextbooks/textbooks/399
  Ref: Lilja, David J. (2016). Linear Regression Using R: An Introduction to Data Modeling. University of Minnesota Libraries Publishing. Retrieved from the University of Minnesota Digital Conservancy, http://hdl.handle.net/11299/189222.
- https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch18.pdf
- https://onlinestatbook.com/2/regression/regression.html
- http://peopleanalytics-regression-book.org

Websites

- https://rhandbook.wordpress.com/tag/regression/
- https://blog.minitab.com/en/how-to-choose-the-best-regression-model
- https://online.stat.psu.edu/stat501/lesson/1
- https://machinelearningmastery.com/regression-metrics-for-machine-learning/
- https://rpubs.com/ryankelly/21379
- https://dataaspirant.com/naive-bayes-classifier-machine-learning/

Modules	
Prediction: Simple Linear Regression	
1.1	Understanding relationship between variables: Covariance, Correlation
1.2	Model and its evaluation
1.3	Testing assumptions of the linear regression algorithm
Prediction: Multiple Linear Regression	
2.1	Interpreting model outputs
2.2	Polynomial regression, Regression with interaction terms
2.3	Data transformations, Handling categorical predictors

2.4	Stepwise regression, Multicollinearity, Error metrics
2.5	Case studies
Classification: Logistic Regression and Naïve Bayes	
3.1	Logistic Regression: Intuition for Maximum Likelihood Estimation and Gradient Descent
3.2	Model output interpretation
3.3	Case study
3.4	Naïve Bayes Classifier: Understanding the naïve assumption, Calculating probabilities
3.5	Case study
3.6	Error/Performance metrics: Confusion Matrix – Precision, Recall/Sensitivity, F1 Measure, Specificity, Accuracy
3.7	Visualizing performance tradeoffs: ROC, AUC, Gains chart, Lift chart

L	T	P
4		

COURSE NAME: Basics of Linear Algebra **COURSE CONTACT HOURS: 4**
COURSE CREDIT HOURS: 4 **COURSE CODE: BTECH MAT 104**

COURSE DESCRIPTION:

This course offers a comprehensive exploration of fundamental concepts in linear algebra. Students will delve into the essential principles of vector spaces, matrix algebra, linear transformations, eigenvalues, and eigenvectors. Emphasizing both theoretical understanding and practical applications, the course equips students with a solid foundation in linear algebra, a critical component for further studies in computer science, machine learning, and various other disciplines.

PRE-REQUISITES: CALCULUS – 1

COURSE OBJECTIVES AND COURSE OUTCOMES

COURSE OBJECTIVES

The objectives of this course are to use and understand matrices. These methods will be used in subsequent courses in algorithms, data science, and machine learning. Students will learn certain key computations such as:

- Solve linear systems Ax = b by elimination.
- Solve problems involving systems of linear equations.
- Solve problems involving matrices and use matrix properties.
- Solve problems using determinants.
- Solve problems using vector spaces and subspaces.
- Calculate eigenvalues and eigenvectors of a matrix.
- Find least squares approximations of a data set and solve problems involving orthogonality.

In addition, students will learn the ideas behind these computations.

COURSE OUTCOMES

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

- Demonstrate a deep understanding of foundational concepts in linear algebra, including vector spaces, matrices, and linear transformations, enabling the application of abstract principles to solve real-world problems (CO1)

- Apply matrix operations, such as multiplication, inversion, and solving linear systems, with precision and efficiency (CO2)
- Utilize geometric interpretations to visually represent and comprehend vector spaces, matrices, and transformations, fostering an intuitive grasp of abstract algebraic concepts and their geometric implications (CO3)
- Analyze and interpret eigenvalues and eigenvectors, recognizing their significance in various applications (CO4)
- Apply linear algebra principles to computer science scenarios, demonstrating competence in utilizing mathematical concepts for computer graphics, data analysis, and machine learning applications (CO5)
- Communicate mathematical concepts and solutions clearly and coherently, both in written and oral formats, showcasing the ability to articulate complex linear algebraic ideas to diverse audiences (CO6)

LEARNING RESOURCES

Required Text Reading

- David C. Lay, Steven R. Lay, and Judi J. McDonald, “Linear Algebra and its Applications”, 5th Edition, Pearson Education Inc., 2016.

BOOKS

- Gilbert Strang, “Introduction to Linear Algebra,” 5th Edition, Wellesley-Cambridge Press, 2016.
  A course based on this book is https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/ and YouTube lectures are at https://www.youtube.com/playlist?list=PL221E2BBF13BECF6C
- Gilbert Strang, “Linear Algebra and Its Applications,” 4th Edition, Cengage Learning, 2005.
- Gene H. Golub and Charles F. Van Loan, “Matrix Computations,” 4th Edition, Johns Hopkins University Press, 2013.
- Roger A. Horn and Charles R. Johnson, “Matrix Analysis,” 2nd Edition, Cambridge University Press, 2012.
- Felix R. Gantmacher, “Matrix Theory,” Vol. 1, Chelsea Publishing Company, 2000.
- Garrett Birkhoff and Saunders Mac Lane, “A Survey of Modern Algebra,” 4th Edition, Macmillan Publishing Company, 1977.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, “Deep Learning,” MIT Press, 2016.

COURSE CONTENT

Unit	Modules
1	**Linear equations in linear algebra**
2	**Matrix algebra**
3	**Determinants**
4	**Vector spaces**
5	**Eigenvalues and eigenvectors**
6	**Orthogonality and least squares**

L	T	P
1		2

COURSE NAME: INTRODUCTION TO COMPUTER SCIENCE AND PROGRAMMING 2 **CONTACT HOURS: 5**
COURSE CREDIT HOURS: 3 **COURSE CODE: BTECH CAP 104**

COURSE DESCRIPTION:

This course is a successor to Introduction to Computer Science and Programming Using Python – 1. Topics covered in this course include the following: higher order functions, exceptions as a control flow mechanism, classes and object-oriented programming, encapsulation and information hiding, algorithmic and computational complexity, asymptotic notation, search and sorting algorithms, hash tables.
Computational thinking is the process of analyzing a problem, and then designing and expressing its solution in such a way that a computer can effectively solve the problem. Computational thinking includes a number of characteristics, such as breaking a problem into small and repetitive ordered steps, logically ordering and analyzing data, and creating solutions that can be effectively implemented as programs running on a computer.
The aim of this course is to take students with basic programming skills to a point where they can solve more complex problems, taking program efficiency into consideration. Besides learning to debug programs, students will also learn about object-oriented programming and some simple algorithms and data structures.

PRE-REQUISITES: Introduction to Computer Science and Programming Using Python – 1

COURSE OBJECTIVES AND COURSE OUTCOMES

The objectives of this course are:

- Apply computational thinking concepts, including sequential logic and problem-solving skills.
- Develop computer solutions for moderately complex problems using effective programming techniques.
- Implement solutions in Python, utilizing appropriate classes, data structures, and algorithms.
- Model real-life problems as computational challenges, demonstrating abstraction and problem translation skills.

COURSE OUTCOMES

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

1. CO1: Apply basic concepts of computational thinking, including sequential logic, abstractions, conceptualization and problem-solving
2. CO2: Develop computer solutions to problems of moderate complexity
3. CO3: Implement solutions using suitable classes and data structures and algorithms in Python
4. CO4: Model real-life problems as computational problems

LEARNING RESOURCES

Required Resources

- John Guttag, "Introduction to Computation and Programming Using Python: With Application to Understanding Data," Second Edition, PHI, 2016.

Recommended Resources:

- Donald Knuth, "The art of computer programming", Volumes 1, Fundamental Algorithms, Addison-Wesley, 1968.
- Mark Lutz, "Learning Python", 5th Edition, O'Reilley, 2013.

COURSE CONTENT

Sr.No	**Modules/ Units**
1	**Testing and debugging**
	Testing Debugging
2	**Exceptions and assertions**

	Handling exceptions Exceptions as a control flow mechanism Assertions
3	**Classes and object-oriented programming**
	Abstract data types and classes Inheritance Encapsulation and information hiding
4	**A simplistic introduction to algorithmic complexity**
	Thinking about computational complexity Asymptotic notation Some important complexity classes
5	**Some simple algorithms and data structures**
	Search algorithms Sorting algorithms Hash tables

L	T	P
2		

COURSE NAME: MODEL THINKING **COURSE CONTACT HOURS: 2**
COURSE CREDIT HOURS: 2 **COURSE CODE: BTECH LA 102**

COURSE DESCRIPTION:

Welcome to the "Model Thinking" course, where we delve into the crucial skill set required by successful engineers: the ability to analyse complex environments through the lens of simple models. Engineers can adeptly organize information and draw efficient inferences from available data using these models. However, using appropriate models requires a detailed understanding of various frameworks and existing models from a variety of disciplines. Those who consistently engage in model construction and thinking tend to outperform their peers.

Throughout this course, we will immerse you in several significant model families sourced from diverse disciplines. Our approach involves not only training you to utilize these models for problem-solving but also imparting the skills necessary to construct models independently. We begin by exploring basic guessing frameworks, progressing to in-depth studies of Simulation, Bayesian, Diffusion, Game Theoretical, Markovian, and Graph models.

By mastering the content of this class, you will establish a solid foundation for subsequent courses that delve more quantitatively into data analysis. The knowledge gained here will provide you with a substantial advantage as you progress into more advanced academic pursuits.

I take quite a bit of inspiration, and content from a course of the same name from Coursera (https://www.coursera.org/learn/model-thinking). However, I will actually restructure it for engineers by adding different use cases, content and exercises.

The course requires just high school mathematics. We will have 3 quizzes and two exams (mid-term and final).

PRE-REQUISITES: None.

However, entry-level proficiency in any programming language is ideal.

COURSE OBJECTIVES AND COURSE OUTCOMES

COURSE OBJECTIVES

Students who complete this course are prepared to transform a complex qualitative-sounding problem into a solvable quantitative model.
The objectives of this course are:

- To study how models are constructed for complex problems in various disciplines
- To learn some powerful models
- To apply models and solve problems

COURSE OUTCOMES

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

1. CO1: Start thinking about problems using models
2. CO2: Gain confidence with the jargon used in various model families and understand the underlying conceptual and mathematical details
3. CO3: Build sufficient mastery to build a model using a programming language of choice

1. Learning Resources

Required Text Reading

- None

BOOKS

- Model Thinking Course in Coursera (https://www.coursera.org/learn/model-thinking). Not all modules are required. But, we will specify specific reading for each class
- As this course requires studying diverse material, I shall provide detailed online class notes for each class along with recommended references

Modules	
Simulations	
1.1	Guessing intelligently: How do we go from simple to complex, known to unknown to guess intelligently? A few fun facts about numbers
1.2	Understanding Monte Carlo simulations and applying them to solve real world problems.

1.3	Introduction to simulated annealing; an update to Montecarlo methods for efficient search.
Graph Models	
2.1	Converting context into a matrix (transition probability matrix), properties of Markov matrices and their applications across disciplines.
2.2	Then, we learn some advanced concepts of probabilistic components and use them as foundations to Bayesian thinking. We will learn how to construct Bayesian nets, and infer from them.
2.3	Representing connected events as graphs (Markovian data as a graph). Graph based problem solving, page rank algorithms
2.4	Case studies
Handling non-linearities and exponential behavior	
3.1	We study taylor series, fourier transforms and exponential approximations to understand non-linearity
3.2	We end the course with diffusion models which are applicable to engineering, healthcare and social phenomenon. We will understand the difference between traditional engineering models and the data models

L	T	P
		1

COURSE NAME: DATA ANALYST PROJECT **COURSE CONTACT HOURS: 2**
COURSE CREDIT HOURS: 1 **COURSE CODE: BTECH CPT 106**

COURSE DESCRIPTION:

This capstone course is designed for first-year B.Tech students with a focus on data analytics. It combines theoretical knowledge with practical applications, using real-world data sets to solve relevant problems. Throughout the course, students will engage in various projects and case studies to develop their analytical skills, learning to use industry-standard tools and techniques to extract, process, and interpret data. The course culminates in a capstone project that requires students to apply their learned skills to a complex data analysis challenge, reflecting both individual learning and teamwork..

COURSE AIMS, LEARNING OBJECTIVES, AND GRADUATE ATTRIBUTES

The objectives of this course are:

- Understand Fundamentals: To equip students with a fundamental understanding of data analytics, including statistical methods, data manipulation, and visualization techniques.
- Develop Analytical Skills: To enhance students' ability to think critically and analytically, enabling them to draw meaningful insights from raw data.
- Tool Proficiency: To provide hands-on experience with industry-standard data analytics tools and software, such as Python, R, SQL, and Tableau.
- Problem Solving: To develop problem-solving skills through structured projects and case studies that mimic real-world data analysis scenarios.
- Collaboration and Communication: To foster teamwork and improve communication skills, preparing students to effectively present data-driven findings to varied audiences.

COURSE OUTCOMES (COS)

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

CO1: Demonstrate a thorough understanding of statistical concepts and data analysis techniques.
CO2: Effectively use various data analytics software and tools to manipulate, analyze, and visualize data.
CO3: Apply critical thinking and analytical skills to interpret data and make informed decisions based on that data.
CO4: Conduct comprehensive data projects from inception to completion, including data collection, analysis, interpretation, and presentation.
CO5: Work collaboratively in teams to tackle complex problems, and communicate technical data insights clearly to both technical and non-technical audiences