BACHELOR OF TECHNOLOGY
FOR ADMISSION BATCH 2023-24
METALLURGICAL ENGINEERING / METALLURGY & MATERIALS ENGINEERING
SECOND YEAR (THIRD SEMESTER)

Sl. No.	Category	Course Code	Course	Contact Hrs. L-T-P	Credit	University Marks	Internal Evaluation
Subject (Theory)							
1	BS	HSBS2001	Mathematics - III	3-0-0	3	100	50
2	PC	MTPC2001	Metallurgical Thermodynamics and Kinetics	3-0-0	3	100	50
3	PC	MTPC2002	Science & Engineering of Materials	3-0-0	3	100	50
4	PC	MTPC2003	Elements of Process Metallurgy	3-0-0	3	100	50
5	PC(ACC)	PCAC2001	Python Programming	3-0-0	2	100	50
		PCAC2002	Data Science Foundations				
		PCAC2003	Web and Application Development				
		PCAC2004	Cloud Computing Foundation				
		PCAC2005	Programming Internet of Things				
		PCAC2006	Robotics : Motion Planning				
		PCAC2007	IT Fundamentals for Cybersecurity - I				
6	HS	HSHS2001	Engineering Economics	3-0-0	2	100	50
		HSHS2002	Organizational Behaviour				
Subject (Sessional / Practical)							
7	PC	MTPC2201	Metallurgical Thermodynamics and Kinetics Lab.	0-0-3	1.5	-	100
8	PC	MTPC2202	Physical Metallurgy Lab.	0-0-3	1.5	-	100
9	PC	MTPC2203	Process Metallurgy Lab.	0-0-3	1.5	-	100
10	PC(ACC)	PCAC2201	Python Programming Lab.	0-0-3	1.5	-	100
		PCAC2202	Data Science Foundations Lab.				
		PCAC2203	Web and Application Development Lab.				
		PCAC2204	Cloud Computing Foundation Lab.				
		PCAC2205	Programming Internet of Things Lab.				
		PCAC2206	Robotics : Motion Planning Lab.				
		PCAC2207	IT Fundamentals for Cybersecurity - I Lab.				
			Total	**18-0-12**	**22**	**600**	700

Note: Click here to view/download the syllabus of the subjects.

HSBS2001 MATHEMATICS-III (3-0-0)

Module 1: Laplace Transforms (8 Hours)
Laplace transforms, inverse transforms, linearity, shifting, transforms of derivatives and integrals, solutionof ODEs, unit step function, Dirac's delta function, differentiation and integration of transforms, convolution, integral equations.

Module 2: Fourier series & Applied PDE's (8 Hours)
Fourier series: Euler's formula, 2π and arbitrary periodic functions, even and odd functions.
Elementary PDE's: Method of separation of variables (simple problems). One dimensional wave equation:solution by separation of variables, One dimensional heat equation: solution by Fourier series.

Module 3: Basic Probability (8 Hours)
Axiomatic definition of probability, Basic properties, conditioning and independence, Random variables (discrete and continuous), probability mass and density functions, cumulative distribution functions, moments of random variables, mean and variance.

Module 4: Probability Distributions (8 Hours)
Discrete Probability distributions: Binomial, Poisson and hyper-geometric distributions. Continuous Probability distributions: exponential, uniform and normal distributions.

Module 5: Applied Statistics (8 Hours)
Random sampling, estimation of parameters, maximum likelihood estimation, confidence intervals. Regression and correlation analysis: fitting of straight lines (method of lest squares), correlation coefficientwith basic properties.

Text Books:

1. Advanced Engineering Mathematics by E. Kreyszig, John Willey & Sons Inc. 10th Edition.
2. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers & Keying Ye, "Probability & Statistics for Engineers & Scientists", Eighth Edition, 2007, Pearson Education Inc., NewDelhi.

Reference Books:

1. Ordinary and Partial Differential equations by J. Sinha Roy and S. Padhy, Kalyani Publishers.
2. Higher Engineering Mathematics by B. V. Ramana, McGraw Hill Education.
3. Engineering Mathematics by Pal and S. Bhunia, Oxford Publication.
   Stochastic Processes, 2nd Edition by Roy D. Yates, Rutgers and David J. Goodman, JohnWiley and Sons, INC.

HSHS2001 ENGINEERING ECONOMICS (3-0-0)

Objectives:

To provide basic concept of micro and macro economics, engineering economics and their application in engineering economy. Further, to develop the ability to account for time value of money using engineering economy factors and formulas.

Module - I (05 hours)

Engineering Economics- Nature, Scope, Basic problems of an economy, Micro Economics and Macro Economics.

Demand - Meaning of demand, Demand function, Law of Demand and its exceptions, Determinants of demand, Elasticity of demand & its measurement (Simple numerical problems to be solved), Demand Forecasting Meaning

Supply-Meaning of supply, Law of supply and its exception, Determinants of supply, Elasticity of supply, Determination of market equilibrium (Simple numerical problems to be solved).

Module - II (O8 hours)

Production - Production function, Laws of returns: Law of variable proportion, Law of returns to scale

Cost and Revenue Concepts - Total Costs, Fixed cost, Variable cost, Total revenue, Average revenue and Marginal revenue, Cost-Output Relationships in the Short Run, and Cost-Output Relationships in the Long Run, Analysis of cost minimization.

Module III (08 hours)

Market - Basic understanding of different market structures, Determination of equilibrium price under perfect competition (Simple numerical problems to be solved), Break Even Analysis-linear approach (Simple numerical problems to be solved).

Module - IV (12 hours)

Time Value of Money- Interest - Simple and compound, nominal and effective rate of interest, Cash flow diagrams, Principles of economic equivalence.

Evaluation of Engineering Projects -Present worth method, Future worth method, Annual worth method, Internal rate of return method, Cost benefit analysis for publicprojects.

Depreciation- Depreciation of capital assert, Causes of depreciation, Methods of calculatingdepreciation - Straight line method, Declining balance method, SOYD method, After tax comparison of project

Module V (06 Hours)

Inflation-Meaning of inflation, types, causes, measures to control inflation.

National Income-Definition, Concepts of national income, Method of measuring national income.

Banking -Commercial bank. Functions of commercial bank, Central bank, Functions of Central Bank.

Books:

1. Principles of Economics by Deviga Vengedasalam and Karaunagaran Madhavan, Oxford
2. Riggs, Bedworth and Randhwa, "Engineering Economics", McGraw Hill Education India
3. C. S. Park, Contemporary Engineering Economics, 6th Edition, Pearson Education, 2015.
4. Engineering Economy by William G.Sullivan, Elin M.Wicks, C. Patric Koelling, Pearson
5. R.Paneer Seelvan, " Engineering Economics", PHI
6. Ahuja,H.L., "Principles of Micro Economics" , S.Chand & Company Ltd
7. Jhingan,M.L., "Macro Economic Theory"
8. Macro Economics by S.P.Gupta, TMH

Course Outcomes of Engineering Economics

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

CO1 Remembering : Define the basic concept of micro and macro economics, engineering economics and their application in engineering economy.

CO2 Understanding : Evaluate numerically the effects of changes in demand and supply on price determination of products and services.

CO3 Analyze : the macroeconomic environment and financial systems of the country and its impact on business, society and enterprise.

CO4 Develop : the ability to account for time value of money using engineering economy factors and formulas.

CO5 Apply: knowledge of mathematics, economics and engineering principles to solve engineering problems and to analyze decision alternatives in engineering projects considering upon depreciation, taxes and inflation.

MTPC2001 METALLURGICAL THERMODYNAMICS AND KINETICS (3-0-0)

Course Objective:
To understand thedifferent laws and principles of thermodynamics and their applications in metallurgical operation.

Module I
Importance of Thermodynamics, definition of thermodynamic terms; concept of states, simple equilibrium. Equation of states, extensive and intensive properties, homogeneous and heterogeneous systems. Phase diagram of a single component system. Internal energy, heat capacity, enthalpy, isothermal, and adiabatic processes.

Module-II
Second law of thermodynamics, entropy, degree of reversibility and irreversibility, criteria of equilibrium, auxiliary functions, combined statements, Maxwell's relations, transformation formula, Gibbs-Helmoltz equation. Concept of Third law of thermodynamics, temperature dependence of entropy, statistical interpretation of entropy, Debye and Einstein concept of heat capacity, relation between Cp and Cv , consequences of third law.

Module III
Fugacity, activity, equilibrium constant, use of S-functions, controlled atmospheres, homogeneous and heterogeneous equilibria. Ellingham – Richardson diagrams, phase stability diagrams.
Solutions: partial molal quantities, ideal and non-ideal solutions, Henry's law, Gibbs – Duhem equation, regular solution, quasi-chemical approach to solution, statistical treatment. One weight percentage standard state, chemical potential, phase relations and phase rule – its applications.

Module IV
Free energy – composition diagrams for binary alloy systems, determination of liquidus, solidus and solvus lines. Effect of pressure on phase transformation and phase equilibria. Thermodynamics of electrochemical cells, solid electrolytes. Thermodynamics of point defects in solids.

Module V
Introduction to metallurgical kinetics: heterogeneous reaction kinetics: gas-solid, solid – liquid, liquid – liquid and solid-solid systems. Empirical and semi-empirical kinetics, concept of Johnson –Mehl equation, thermal analysis.

Course Outcome:
CO1: To understand the concept and importance of thermodynamics
CO2: To understand and apply the laws of thermodynamics
CO3: To apply the knowledge of thermodynamics in the real engineering world.
CO4: To interpret and apply the data of thermodynamics in the Metallurgical Engineering processes.
CO5: To identify and recommend the optimum operational parameters to be employed in significant Metallurgical Engineering processes.

Text Books:

1. GaskellD.R., Introduction to the Thermodynamics of Materials; Taylor and Francis.
2. GhoshA., Textbook of Materials and Metallurgical Thermodynamics; Prentice Hall of India Pvt. Ltd.

Reference Books:

1. Bose S.K. and Roy S.K., Principles of Metallurgical Thermodynamics; University Press
2. TupkaryR.H.,Essentials of Metallurgical Thermodynamics; Khanna Book Publishing

MTPC2002 SCIENCE & ENGINEERING OF MATERIALS (3-0-0)

MODULE-I (08 hrs)

Bonding in solid*: Ioning, Covalent and metallic bonding; Bonding forces and energy, Secondary bonding*
Crystal structure*: Space lattices and Bravais lattices, Miller indices of planes and directions, slip planes and Slip directions, Stereographic projections.*
Selected crystal structures*: Pure metals; Diamond and graphite, coordination in ionic crystals, AB type compounds, Silica, Alumina, Complex oxides, Silicate, Inorganic glass; Network structure in glasses, polymeric structures: Thermoplastics, Elastomers, crystallinity in polymers.*

MODULE-II (08 hrs)

Principles of Alloy theory: Primary substitutional solid solution, Interstitial solid solution, types of intermediate phase, Ordered-Disordered Phenomena. Hume Rothery Rules, Intermetallic compounds, Normal valency compounds, Electron compounds, Interstitial compounds.
Imperfections: Point defects, Vacancies, Interstitial defects, Edge and Screw Dislocations, Burger's vector, Crystallization From the melt: Freezing of pure metal, plane front and dendritic solidification at a cooled surface, formation of cast structure, Gas porosity and segregation, directional solidification.

MODULE-III (08 hrs)

Binary Phase Diagrams; Isomorphous, Eutectic, Peritectic, Eutectoid, Monotectic and Syntectic system, Phase rule and Lever rule. Iron-Cementite Equilibrium diagrams and its application, Plane carbon and alloy steel, Industrial application of steels.

MODULE-IV (08 hrs)

Diffusion: Fick's First and Second law of diffusion, atomic model of Diffusion, Grain boundary, Surface and thermal diffusion Kirkendall effect, Industrial diffusion.

MODULE-V (08 hrs)

Nucleation: Homogeneous and Heterogeneous nucleation, Kinetics of nucleation, growth and overall transformation kinetics.

Books:

• Introduction to Physical Metallurgy by Avner, Tata McGraw Hill
• Materials Science and Engineering by W.D.Callister, Wiley and Sons Inc.
• Physical Metallurgy: Principles and Practice by Ragahvan, PHI.

MTPC2003 ELEMENTS OF PROCESS METALLURGY (3-0-0)

MODULE-I (08 hrs)

Overview of Extractive Metallurgy processes; Pyro-metallurgy, Hydrometallurgy and Electrometallurgy; Thermodynamic and Kinetic Principles of metal extraction; Ellingham diagrams, Calcinations; Roasting; Predominance Area Diagram, Roasting Practices, Smelting, Formation and function of slag and their calculations,

MODULE-II (08 hrs)

Metallo-thremic and carbothermic reduction of oxides, Smelting Furnaces, Matte Smelting, Pyro metallurgical processes using vacuum Hydrometallurgy: Leaching; Theory of Leaching; Role of oxygen in leaching operation; Bacterial and microbial leaching; Contact reduction of metals in aqueous solutions;

MODULE-III (08 hrs)

Gaseous reduction of metals in aqueous solutions; Ion exchange, Solvent Extraction and Electrolysis, Electrometallurgy: laws of electrolysis, electrolyte Structure of solvent media; Electrolysis of aqueous solution; Electrolysis of fused salts; Cell design; Electro refining

MODULE-IV (08 hrs)

Halide Metallurgy and Halogenation., Basic approaches of refining, preparation of pure compounds; Purification of crude metals produced in bulk

MODULE-V (08 hrs)

Concept of activity, chemical potential, fugacity, real and idle solution, and the significance in metal extraction, Numerical problems relevant to Pyro, Hydro and Electrometallurgical processes

Books:

[1] Principles of Extractive Metallurgy: A. Ghosh & H.S. Ray, IIN Publications, Kolkata 1984

[2] Principles of Extractive Metallurgy: Rosenquist, T., McGrawhill - Kogakusha International – 1983 [3] Mineral Processing and Extractive Metallurgy by Corby G. Anderson (Editor), Robert C. Dunne (Editor), John L. Uhrie (Editor)

[4] Metallurgy a Brief Outline of the Modern Processes for Extracting the More Important Metals by W. Borchers

MTPC2201 METALLURGICAL THERMODYNAMICS AND KINETICS LAB. (0-0-3)

1. Introduction to terms, properties and safety related to thermodynamics
2. Isothermal kinetic study of limestone decomposition.
3. Devolatilization kinetics of coal.
4. Study the Kinetics of oxidation of copper in atmospheric air by the weight gain method.
5. Determination of standard free energy, enthalpy and entropy of formation of calcium carbonate through pressure measurement technique
6. Study the kinetics of roasting of ZnS by weight loss method.
7. Determination of partial molal volume of each component in a solution of water and ethanol.
8. Study the effect of time and temperature on reduction of iron ore pellet.
9. Dilatometric study of given Al plate sample
10. Thermal analysis of 0. 2% C steel using DSC – TG technique to study Microstructural changes with temperature

Course Outcome:

CO1: To acquire the knowledge for the basic operations and safety in thermodynamics
CO2: To analyze the kinetics of reactions of limestone, coal and ZnS
CO3: To acquire the knowledge of free energy, enthalpy and entropy concept in thermodynamics
CO4: To analyze the properties of solutions
CO5: To evaluate and visualize the concept of thermal analysis

MTPC2202 PHYSICAL METALLURGY LAB. (0-0-3)

List of experiments

1. Study of metallurgical microscope
2. Metallographic preparation of metals and alloys
3. Microscopic examination of steels and non-ferrous metals, interpretation of microstructures, Microstructures of heat-treated steels
4. Microscopic examination of cast, wrought and welded structures, defects and failure in components
5. Comparison of microstructures of cold worked and annealed samples

MTPC2203 PROCESS METALLURGY LAB. (0-0-3)

List of experiments

1. To determine and analyze the size distribution of a fixed granular solid by using a test
2. sieve stack and a vibratory shaker.
3. Crushing of ore/coal in the jaw crusher and average size determination by sieving.
4. To study the jaw crusher and determine the actual capacity and reduction ratio, and verification of Rittinger's law of crushing.
5. Crushing of ore/coal in a roll crusher and average size determination by sieving.
6. To study the effect of grinding with grinding time in ball mill.
7. To separate a mixture of two minerals of different densities by gravity
8. concentration using Wilfley Table and determine the weight and density of each fraction of the products
9. Beneficiation of ore pulp mix using floatation cell.
10. Study of magnetic separator and effect of field on the efficiency of the process

PCAR 2001 PYTHON PROGRAMMING (3-0-0)

OVERALL COURSE OBJECTIVES: The objective of this course is to provide learners with a comprehensive understanding of Python, from basic programming to handling complex data structures and accessing web data. By the end, learners should be proficient in Python and be able to use their skills to extract, parse, and analyze data. Moreover, they should be equipped to take on further advanced programming courses.

Module 1: Programming for Everybody (Getting Started with Python) [19 Hours]

This course aims to teach everyone the basics of programming computers using Python. We cover the basics of how one constructs a program from a series of simple instructions in Python. The course has no prerequisites and avoids all but the simplest mathematics. Anyone with moderate computer experience should be able to master the materials in this course. This course will cover Chapters 1-5 of the textbook "Python for Everybody". Once a student completes this course, they will be ready to take more advanced programming courses. This course covers Python 3.

Sub-Topics

Installing Python
Python as a Language
Eben Upton and the RaspBerry Pi
Variables and Expressions
Conditional Code
Conditional Statements
Loops and Iteration

Formative Assessments:

5 quizzes, 1 peer-review assignment, and 7 coding/lab assignments.

Module 2: Python Data Structures [19 Hours]

This course will introduce the core data structures of the Python programming language. We will move past the basics of procedural programming and explore how we can use the Python built-in data structures such as lists, dictionaries, and tuples to perform increasingly complex data analysis. This course will cover Chapters 6-10 of the textbook "Python for Everybody". This course covers Python 3.

Sub-Topics

Strings
Files
Lists
Dictionaries
Tuples

Formative Assessments:

5 quizzes, 1 peer-review assignment, and 7 coding/lab assignments.

Module 3: Using Python to Access Web Data [19 Hours]

This course will show how one can treat the Internet as a source of data. We will scrape, parse, and read web data as well as access data using web APIs. We will work with HTML, XML, and JSON data formats in Python. This course will cover Chapters 11-13 of the textbook "Python for Everybody". To succeed in this course, you should be familiar with the material covered in Chapters 1-10 of the textbook and the first two courses in this specialization. These topics include variables and expressions, conditional execution (loops, branching, and try/except), functions, Python data structures (strings, lists, dictionaries, and tuples), and manipulating files. This course covers Python 3.

Sub-Topics

Regular Expressions
Networks and Sockets

Programs that Surf the Web
Web Services and XML
JSON and the REST Architecture

Formative Assessments:
5 quizzes and 8 coding/lab assignments

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Demonstrate understanding of basic programming concepts in Python, including constructing simple programs.
2. Apply gained Python proficiency to pursue more advanced programming courses.
3. Evaluate and use Python's core data structures such as lists, dictionaries, and tuples for sophisticated data analysis.
4. Extract and interpret data from the internet using Python’s web scraping tools and APIs.
5. Interpret and manipulate web data, specifically HTML, XML, and JSON, using Python.
6. Synthesize various Python concepts, such as handling different data structures and manipulation of web data, to solve complex problems.

PCAC2002 DATA SCIENCE FOUNDATIONS (3-0-0)

OVERALL COURSE OBJECTIVES: To enhance students' aptitude in implementing scalable data science platforms, and understanding big data landscape with a focus on using statistical measures, data visualization, advanced tools, and specific processes that aid in detecting data trends, minimizing inconsistencies, and improving overall data analysis.

Module 1: Introduction to Data Science in Python [35 Hours]

This course will introduce the learner to the basics of the Python programming environment, including fundamental Python programming techniques such as lambdas, reading and manipulating csv files, and the numpy library. The course will introduce data manipulation and cleaning techniques using the popular Python pandas data science library and introduce the abstraction of the Series and DataFrame as the central data structures for data analysis, along with tutorials on how to use functions such as group by, merge, and pivot tables effectively. By the end of this course, students will be able to take tabular data, clean it, manipulate it, and run basic inferential statistical analyses.

Sub-Topic

Fundamentals of Data Manipulation with Python
Data Processing with Pandas
Answering Questions with Messy Data

Formative Assessments:

4 quizzes and 9 coding/lab assignments.

Module 2: Introduction to Big Data [17 Hours]

This course provides an introduction to the Big Data landscape for beginners interested in data science. It includes an overview of key concepts behind big data problems, applications, and systems. The course offers familiarity with the Hadoop framework that simplifies big data analysis, making it more accessible. It covers the characteristics of Big Data, the process of structuring analysis, identification of big data problems, the architectural components, and programming models for scalable big data analysis. It also explores the core Hadoop stack components including the YARN resource and job management system, the HDFS file system, and the MapReduce programming model. Installations and virtual machine operations are required for hands-on assignments. Prior programming experience is not necessary.

Sub-Topic

Big Data: Why and Where
Characteristics of Big Data and Dimensions of Scalability
Data Science: Getting Value out of Big Data
Foundations for Big Data Systems and Programming
Systems: Getting Started with Hadoop

Formative Assessments:

6 quizzes and 1 peer-review assignment.

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Understand and apply basic statistical measures to identify patterns within large sets of data,
2. Develop proficiency in recognizing various data characteristics, patterns, trends, deviations or inconsistencies, and potential outliers.
3. Employ techniques for dealing with big data like dimension reduction and feature selection methods.
4. Leverage advanced tools and charting libraries to improve the efficiency of big data analysis with partitioning and parallel analysis.
5. Visualize data using 2D and 3D formats achieving a better understanding and interpretation.
6. Get value out of Big Data following a specific 5-step process to structure your analysis.

PCAC2003 WEB AND APPLICATION DEVELOPMENT (3-0-0)

OVERALL COURSE OBJECTIVES: To enable learners to apply HTML5, CSS, Javascript, Git, GitHub, React, Node.js, and Express effectively in creating dynamic and interactive websites and web applications, understand and implement front-end and back-end development practices, effectively use version control for collaboration and demonstrate competencies in widely-used web technologies and server-side frameworks.

Module 1: Introduction to Web Development with HTML, CSS, JavaScript [13 Hours]

This starter course is designed for individuals aiming to become Web Developers, offering an introduction to the roles of front-end, back-end, and full-stack developers in development projects. It also familiarizes learners with the terminology and skills essential for a web development career. The focus is given to the languages needed for website or application development with a comprehensive understanding of HTML and CSS for creating the structure and style of websites. JavaScript is introduced to enable dynamic page features like interactive forms, dynamic content modification, and sophisticated menu systems. On completing this course, learners will be able to create a basic structure for a website, format and layout for web applications, enhance websites with rich, interactive applications, increase user interactivity and experience, and provide their websites with a unique appeal. Hands-on labs provide practical application opportunities, and a final portfolio-worthy project involves creating a webpage to showcase the skills learned.

Sub-Topic

Introduction to Application Development
CSS Overview & HTML5 Elements
HTML Overview
JavaScript Programming for Web Applications

Formative Assessments:

4 graded quizzes and 1 Peer-review assignment.

Module 2: Getting Started with Git and GitHub [18 Hours]

This self-paced introductory course provides an in-depth understanding of Git and GitHub, essential tools for collaboration and social coding in modern software engineering and DevOps culture. Starting with Git and GitHub fundamentals, it covers key Git concepts such as branching and repositories, along with the use of Git commands. The course includes hands-on labs, augmenting understanding of Git concepts including forking, cloning, and merging workflows, and fostering team productivity on GitHub. It concludes with a final project that allows students to begin building their portfolio with a public/open-source GitHub project, thus demonstrating their Git and GitHub skills and providing a valuable addition to their resume. All activities are browser-based, negating any need for specialized software installation on the learner's computer.

Sub-Topic

Git and GitHub Fundamentals
Using Git Commands and Managing GitHub Projects
Cloning and Forking GitHub Projects

Formative Assessments:

2 graded quizzes and 1 Peer-review assignment.

Module 3: Developing Front-End Apps with React [14 Hours]

This course provides comprehensive instruction on React, a popular framework for web and front-end application development. The curriculum includes building rich front-end applications with React and ES6, connecting React components using data and state, and writing advanced React components using Hooks and Redux. Learners will gain access to the React web framework UI library and learn to run rich React applications, modify their properties and states, and connect to an external server from a React page. The course also introduces various testing tools to verify components without manual checking. Hands-on labs and a final portfolio-worthy project form part of the course, demonstrating

learners' acquired React skills. This course is beneficial for those looking to further their IT career as front-end or full-stack developers. Prior knowledge of HTML, CSS, JavaScript, and Git/GitHub is required.

Sub-Topic

Advanced React
Building Rich Front-End Applications with React and ES6
React Components
Introduction to TypeScript
Passing Data and States Between Components

Formative Assessments:

3 graded quizzes and 1 Peer-review assignment.

Module 4: Developing Back-End Apps with Node js and Express [12 Hours]

This course primarily focuses on Node.js and Express, two popular web technologies. Node.js, the most commonly used server-side technology, and Express, the most prevalent server-side web framework, are vital for developing modern web applications. In this course, you will concentrate on crafting applications using asynchronous callbacks and promises, creating REST APIs, and performing CRUD operations. You will also learn to implement authentication and session management. Ample hands-on labs provide practical experience, and a final project allows you to demonstrate your Node.js skills and add to your portfolio. This course equips you to thrive as a back-end or full-stack developer and is perfect for IT professionals aspiring for career advancement, new graduates looking to refine their server-side skills, and those managing cloud-centric projects. Prerequisites include knowledge of JavaScript and Git.

Sub-Topics

Introduction to Server-Side JavaScript
Asynchronous I/O with callback programming
Express Web Application Framework

Formative Assessments:

3 graded quizzes and 1 Peer-review assignment.

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Demonstrate the fundamentals of HTML5, CSS, and JavaScript to create dynamic websites and web applications.
2. Utilize Git and GitHub for version control, collaboration, and social coding effectively in software engineering and DevOps practices.
3. Leverage React and ES6 to construct rich and interactive front-end applications with features like Hooks and Redux.
4. Design and manipulate dynamic user interfaces through React components, their properties, and states.
5. Develop back-end applications using Node.js and Express with features like asynchronous callbacks, REST APIs, CRUD operations, and session management.
6. Exhibit proficiency in server-side technologies, focusing on most popular server-side web framework- Express.

PCAR2004 CLOUD COMPUTING FOUNDATIONS (3-0-0)

OVERALL COURSE OBJECTIVES: The objective of this course is to enable learners to understand and analyze the fundamentals of cloud computing, its architecture, and emerging trends, and apply distributed computing concepts practically using relevant programming tools in the field, ultimately preparing them for potential career paths in cloud-based roles.

Module 1: Introduction to Cloud Computing [24 Hours]

This self-paced introductory course sets learners on a journey through the essentials of cloud computing. Appropriate for students, business professionals, and those considering a career switch, it covers vital characteristics of cloud computing, emergent technologies, service models including IaaS, PaaS, and SaaS, and deployment models like Public, Private, and Hybrid. Learners will explore the offerings of major cloud service providers, study case scenarios, and delve into topics like cloud adoption, blockchain, analytics, AI, and cloud computing architecture components. They will also become familiar with different types of cloud storage options and emergent cloud trends. The course concludes with students deploying an application to the cloud using serverless architecture as a final project, thus providing them with a beneficial portfolio addition.

Sub-Topics

Overview of Cloud Computing
Cloud Computing Models
Components of Cloud Computing
Emergent Trends and Practices
Cloud Security and Monitoring, Case Studies, and Jobs
Final Project and assignment

Formative Assessments:

5 quizzes and 1 peer-review assignment.

Module 2: Cloud Computing Concepts, Part 1 [23 Hours]

Cloud computing systems today, whether open-source or used inside companies, are built using a common set of core techniques, algorithms, and design philosophies – all centered around distributed systems. Learn about such fundamental distributed computing "concepts" for cloud computing. Some of these concepts include: clouds, MapReduce, key-value/NoSQL stores, classical distributed algorithms, widely-used distributed algorithms, scalability, trending areas, and much, much more! Know how these systems work from the inside out. Get your hands dirty using these concepts with provided homework exercises. In the programming assignments, implement some of these concepts in template code (programs) provided in the C++ programming language. Prior experience with C++ is required. The course also features interviews with leading researchers and managers, from both industry and academia.

Sub-Topics

Gossip, Membership, and Grids
P2P Systems
Key-Value Stores, Time, and Ordering
Classical Distributed Algorithms

Formative Assessments:

6 quizzes, and 1 coding/lab assignments.

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Understand the essential features and various service models of cloud computing along with the offerings of prominent market players.
2. Analyze different components of cloud computing architecture such as data centers, virtual machines, containers, and cloud storage options.
3. Demonstrate knowledge of emergent cloud trends such as DevOps, Hybrid and MultiCloud, and cloud security and monitoring.

4. Evaluate the applications of cloud computing in areas like blockchain, analytics, AI, and job roles in this field.
5. Develop competence in distributed computing concepts such as MapReduce, key-value/NoSQL stores, and scalability techniques used in cloud computing.
6. Apply these concepts practically to build or manipulate cloud systems using programming languages like C++.

PCAC2005 PROGRAMMING INTERNET OF THINGS (3-0-0)

OVERALL COURSE OBJECTIVES: To empower students with a comprehensive understanding of IoT and Embedded Systems, Arduino and Raspberry Pi platforms, and C and Python programming. This will enable them to create innovative IoT designs and products and understand how these devices interact with the physical world. They will also learn debugging techniques and network protocols essential for embedded systems.

Module 1: Introduction to the Internet of Things and Embedded Systems [12 Hours]

This course explores the significant role of the "Internet of Things" (IoT) in the modern world and its future trends. It defines what IoT and embedded systems are, describes their impact on society, and enumerates their components. The lessons cover hardware and software interactions in an IoT device and the role of an operating system in supporting this software. The course highlights key components of networking, including an understanding of how to connect devices to the Internet, the structure of the Internet, and the meaning of a "network protocol". It also explains Mobile Ad-Hoc Networks (MANETs) in relation to IoT. While beneficial, this course does not include discussion forums.

Sub-Topic

Embedded Systems
Hardware and Software
Networking and the Internet
What Is the Internet of Things (IoT)?

Formative Assessments:

4 quizzes and 4 peer-review assignments.

Module 2: The Arduino Platform and C Programming [13 Hours]

This course provides in-depth knowledge about the Arduino platform, including the physical board, libraries, and the integrated development environment (IDE). It explores the role and usage of shields and touches on programming the Arduino using C code. The lessons delve into elements like reading board schematics, installing the Arduino IDE, understanding the significance of libraries, and running a program. The course provides a comprehensive understanding of C variables, types, common operators, conditionals, loops, functions, and the implications of global variables. Additionally, the course covers the Arduino build process, the role of tools in the IDE, the structure of an Arduino sketch, and accessing pins on the Arduino. It also covers embedded software debugging, common debugging architectures for embedded systems, and the UART Serial communication protocol. The course does not include discussion forums.

Sub-Topic

Arduino Environment
Arduino Programs
C Programming
Basic C Operators
Arduino Sketches

Formative Assessments:

4 quizzes and 4 peer-review assignments.

Module 3: The Raspberry Pi Platform and Python Programming for the Raspberry Pi [19 Hours]

The Raspberry Pi is a small, affordable single-board computer that you will use to design and develop fun and practical IoT devices while learning programming and computer hardware. In addition, you will learn how to set up the Raspberry Pi environment, get a Linux operating system running, and write and execute some basic Python code on the Raspberry Pi. You will also learn how to use Python-based IDE (integrated development environments) for the Raspberry Pi and how to trace and debug Python code on the device.

Sub-Topic

Raspberry Pi Processor

Operating System Benefits
Raspberry Pi Configuration
Navigating the Filesystem
Linux Graphic User Interface
Python on Raspberry Pi

Formative Assessments:
4 quizzes and 4 peer-review assignments.

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Understand and define the key concepts of “Internet of Things” and its impact on society, focusing specifically on design considerations and components of IoT devices.
2. Master the composition and firmware programming of the Arduino development board, as well as the usage of "shields" and libraries.
3. Gain the ability to compile and run a program using C language, understanding variables, types, and operators specifically relevant to Arduino sketches.
4. Acquire knowledge on the Raspberry Pi setup and operation, including executing a Linux operating system.
5. Develop expertise in writing and executing basic Python code on Raspberry Pi, also learning to use Python-based IDEs and debugging Python code.
6. Understand the fundamental aspects of networking, including network protocol, structure of the Internet, and their specific implications in IoT devices.

PCAC2006 ROBOTICS: MOTION PLANNING (3-0-0)

OVERALL COURSE OBJECTIVES: To develop a comprehensive understanding of robotics including aerial flight mechanics, computational motion planning, and their applications in the drone industry; achieve capabilities to build dynamic models, devise controllers, and navigate in complex environments using methods such as graph-based methods and artificial potential fields.

Module 1: Robotics: Aerial Robotics [18 Hours]

This course delves into the creation of agile micro aerial vehicles capable of operating autonomously in cluttered indoor and outdoor environments. It introduces the mechanics of flight and the design of quadrotor flying robots, enabling you to develop dynamic models, derive controllers, and synthesize planners for three-dimensional environments. Faced with the challenges of utilizing noisy sensors for localization and complex, three-dimensional maneuvering. The course presents real-world examples of the potential applications and challenges in the rapidly-growing drone industry. A familiarity with linear algebra, single-variable calculus, and differential equations, along with some experience programming with MATLAB or Octave, is recommended for those planning to take this course.

Sub-Topics

Key Components of Autonomous Flight
Unmanned Aerial Robotics (UAVs) and quadrotors
Design Considerations
Time, Motion, and Trajectories
Axis/Angle Representations for Rotations
Control of Multiple Robots

Formative Assessments:

5 quizzes and 5 Programming assignments.

Module 2: Robotics: Computational Motion Planning [11 Hours]

Robotic systems typically include three components: a mechanism which is capable of exerting forces and torques on the environment, a perception system for sensing the world and a decision and control system which modulates the robot's behavior to achieve the desired ends. In this course we will consider the problem of how a robot decides what to do to achieve its goals. This problem is often referred to as Motion Planning and it has been formulated in various ways to model different situations. You will learn some of the most common approaches to addressing this problem including graph-based methods, randomized planners and artificial potential fields. Throughout the course, we will discuss the aspects of the problem that make planning challenging.

Sub-Topics

Artificial Potential Field Methods
Configuration Space
Collision Detection and Freespace Sampling Methods
Graph-based Plan Methods
Sampling-based Planning Methods
Probabilistic Road Maps

Formative Assessments:

4 quizzes, and 6 coding/lab assignments.

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Understand the mechanics of flight and the design of quadrotor flying robots for operation in 3D environments.
2. Develop dynamic models, derive controllers, and synthesize planners for drone operation.
3. Overcome challenges of using noisy sensors for localization and maneuvering in complex environments.

4. Familiarize with the components of robotic systems: mechanism, perception system, and decision and control system.
5. Grasp common approaches for motion planning in robotics including graph-based methods, randomized planners, and artificial potential fields.
6. Analyze real-world examples of the applications and challenges for the rapidly-growing drone industry.

PCAC2007 IT FUNDAMENTALS FOR CYBERSECURITY – I (3-0-0)

OVERALL COURSE OBJECTIVES: The objective of this course is to equip learners with a comprehensive understanding of Cybersecurity, from foundational knowledge and terminology to practical skills in system operations, role-based security processes, and advanced topics like encryption and compliance standards. This holistic view aims to prepare participants for junior-level analyst roles in the Cybersecurity field, ensuring they are well-versed in both theoretical and practical aspects of cyber defense.

Module 1 : Introduction to Cybersecurity Tools & Cyber Attacks [18 Hours]

This course gives you the background needed to understand basic Cybersecurity. You will learn the history of Cybersecurity, types and motives of cyber attacks to further your knowledge of current threats to organizations and individuals. Key terminology, basic system concepts and tools will be examined as an introduction to the Cybersecurity field. You will learn about critical thinking and its importance to anyone looking to pursue a career in Cybersecurity.

Sub-Topics

A brief overview of types of actors and their motives
An overview of key security concepts
An overview of key security tools
History of Cybersecurity

Formative Assessments:

4 Graded Quizzes

Module 2 : Cybersecurity Roles, Processes & Operating System Security [15 Hours]

This course gives you the background needed to understand basic cybersecurity around people. process and technology. You will understand the key cybersecurity roles within an organization; list key cybersecurity processes and an example of each process; describe the architecture, file systems, and basic commands for multiple operating systems including Windows, Mac/OS, Linux, and Mobile; and also understand the concept of virtualization as it relates to cybersecurity.

Sub-Topics

Authentication and Access Control
Examples & Principles of the CIA Triad
Linux Operating System Security Basics
macOS Security Basics
Overview of Virtualization
People Process & Technology
Windows Operating System Security Basics

Formative Assessments:

6 Graded Quizzes

Module 3 : Cybersecurity Compliance Framework & System Administration [21 Hours]

This course gives you the background needed to understand the key cybersecurity compliance and industry standards. This knowledge will be important for you to learn no matter what cybersecurity role you would like to acquire or have within an organization.

You will learn the basic commands for user and server administration as it relates to security. You will need this skill to be able to understand vulnerabilities within your organizations operating systems.

Sub-Topics

Client System Administration, Endpoint Protection and Patching
Compliance Frameworks and Industry Standards
Cryptography and Compliance Pitfalls
Linux and Encryption: Final Project
Server and User Administration

Formative Assessments:
4 Graded Quizzes

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Understand basic Cybersecurity concepts, gaining foundational knowledge of the Cybersecurity landscape including types, motives of cyber attacks, and the history behind them.
2. Grasp key Cybersecurity terminology and tools, learning essential terms and introductory tools relevant to Cybersecurity, facilitating a deeper understanding of system concepts.
3. Recognize the key roles and typical processes within a Cybersecurity organization, enhancing comprehension of operational security.
4. Develop skills to navigate and manage Windows, MacOS, Linux, and mobile operating systems from a security perspective.
5. Understand and apply cybersecurity compliance standards and protocols to maintain the integrity and security of information systems.
6. Learn fundamental concepts and practices of cryptography and encryption, crucial for protecting information against cyber threats.

PCAC2201 PYTHON PROGRAMMING LAB. (0-0-3)

List of lab/practical assignments:

Module Title	Lab Name	Details
Programming for Everybody (Getting Started with Python)	Python Code Playground	Write any code in the Python playground, referring the three sample files loaded and ready
	Installing Python Screen Shots	Install Python and a programming text editor and write a program that prints one line other than 'hello world', then take two screenshots and upload them. You should use the command line to execute the Python program you wrote in the text editor.
	Write Hello World	Write a program that uses a print function to say 'hello world' as shown in 'Desired Output'.
	Print	Write a program that uses input to prompt a user for their name and then welcomes them
	Input, Float	Write a program that uses input to prompt a user for their name and then welcomes them
	Compute gross pay	Write a program to prompt the user for hours and rate per hour using input to compute gross pay. Pay the hourly rate for the hours up to 40 and 1.5 times the hourly rate for all hours worked above 40 hours. Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75). You should use input to read a string and float() to convert the string to a number.
	Scoring/ Grading	Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error. If the score is between 0.0 and 1.0, print a grade
	Compute gross pay	Write a program to prompt the user for hours and rate per hour using input to compute gross pay.
	Largest/smallest number	Write a program that repeatedly prompts a user for integer numbers until the user enters 'done'. Once 'done' is entered, print out the largest and smallest of the numbers.

Python Data Structures	Find function	Write code using find() and string slicing (see section 6.10) to extract the number at the end of the line below. Convert the extracted value to a floating point number and print it out.
	File processing	Write any code using the three files loaded: "mbox-short.txt", "romeo.txt", and "words.txt".
		Write a program that prompts for a file name, then opens that file and reads through the file, and print the contents of the file in upper case.
		Write a program that prompts for a file name, then opens that file and reads through the file, looking for lines of the form: X-DSPAM-Confidence: 0.8475 Count these lines and extract the floating point values from each of the lines and compute the average of those values and produce an output as shown
	Lists	Open the file romeo.txt and read it line by line. For each line, split the line into a list of words using the split() method.
		Open the file mbox-short.txt and read it line by line. When you find a line that starts with 'From ', find and print out the second word in the line
	Dictionary	Write a program to read through the mbox-short.txt and figure out who has sent the greatest number of mail messages.
	Tuples	Write a program to read through the mbox-short.txt and figure out the distribution by hour of the day for each of the messages.
Using Python to Access Web Data	Installing and Running Python Screen Shots	Install Python and a programming text editor and write a program that prints one line other than 'hello world
	Extracting Data With Regular Expressions	Read through and parse a file with text and numbers. You will extract all the numbers in the file and compute the sum of the numbers.
	Understanding the Request / Response Cycle	Write a Python program to retrieve a web page over a socket and display the headers from the web server.
	Scraping HTML Data with BeautifulSoup	Write a Python program to use urllib to read the HTML from the data files below, and parse the data, extracting numbers and compute the sum of the numbers in the file

	Following Links in HTML Using BeautifulSoup	Write a Python program that will use urllib to read the HTML from the data files below, extract the href= values from the anchor tags, scan for a tag that is in a particular position from the top and follow that link, repeat the process a number of times, and report the last name you find.
	Extracting Data from XML	Write a Python program that will prompt for a URL, read the XML data from that URL using urllib and then parse and extract the comment counts from the XML data, compute the sum of the numbers in the file and enter the sum
	Extracting Data from JSON	The program will prompt for a URL, read the JSON data from that URL using urllib and then parse and extract the comment counts from the JSON data, compute the sum of the numbers in the file.
	Using a Geo Location API	Use a Geo Location lookup API to look up the location of some universities and parse the returned data.

PCAC2202 DATA SCIENCE FOUNDATIONS LAB. (0-0-3)

Module Title	Lab Name	Details
Introduction to Data Science in Python	Fundamentals of Data Manipulation	Find a list of all of all of the names in the following string using regex The dataset file in assets/grades.txt contains a line separated list of people with their grade in a class. Create a regex to generate a list of just those students who received a B in the course. Consider the standard web log file in assets/logdata.txt. Your task is to convert this into a list of dictionaries
	Basic Data Processing with Pandas	Look at 2017 data on immunizations from the CDC. Write a function called proportion_of_education which returns the proportion of children in the dataset who had a mother with the education levels equal to less than high school (<12), high school (12), more than high school but not a college graduate (>12) and college degree. explore the relationship between being fed breastmilk as a child and getting a seasonal influenza vaccine from a healthcare provider. see if there is any evidence of a link between vaccine effectiveness and sex of the child. Calculate the ratio of the number of children who contracted chickenpox but were vaccinated against it (at least one varicella dose) versus those who were vaccinated but did not contract chicken pox. Return results by sex.
	More Data Manipulation with Pandas	Load the energy data from the file assets/Energy Indicators.xls, which is a list of indicators of energy supply and renewable electricity production from the United Nations for the year 2013, and should be put into a DataFrame with the variable name of Energy. Answer questions like: What are the top 15 countries for average GDP over the last 10 years? By how much had the GDP changed over the 10 year span for the country with the 6th largest average GDP?

	Beyond Data Manipulation	Read in a file of metropolitan regions and associated sports teams from assets/wikipedia_data.html and answer some questions about each metropolitan region. calculate the win/loss ratio's correlation with the population of the city it is in for the NHL using 2018 data. calculate the win/loss ratio's correlation with the population of the city it is in for the NBA using 2018 data. calculate the win/loss ratio's correlation with the population of the city it is in for the MLB using 2018 data. calculate the win/loss ratio's correlation with the population of the city it is in for the NFL using 2018 data. explore the hypothesis that given that an area has two sports teams in different sports, those teams will perform the same within their respective sports.
Introduction to Big Data	Understand by Doing: MapReduce	Perform the steps of MapReduce to calculate a count of the number of squares, stars, circles, hearts and triangles in the dataset shown in the picture, following the steps of MapReduce

PCAC2203 WEB AND APPLICATION DEVELOPMENT LAB. (0-0-3)

Module Title	Lab Name	Details
Introduction to Web Development with HTML, CSS, JavaScript	Intro to HTML (JSFiddle)	In this lab, we will explore all the components of an HTML file viz, HTML markup, CSS styling and JavaScript code.
	HTML - Creating a Simple Web Page	In this lab assume you are a web developer who has been approached by the fan club for IBM founder, Thomas J. Watson Sr., to create a web page for them.
	Unit Conversion using HTML5 Structural Elements	In this lab, you will be creating a simple web page Unit Converter application that converts metric units to imperial units. The page will be divided into three sections, each of which will do one of the following: Convert temperature from Celcius to Fahrenheit Convert weight from Kilograms to Pounds Convert distance from Kilometers to Miles
	CSS Basics - Styling Your Web Page	You are a web developer and have been asked by a primary school to develop a web page on the solar system. Objectives After completing this lab, you will be able to: Create a basic webpage Specify the font family and font size for text Use colors to change the appearance of elements Create borders
	JavaScript	JavaScript is a client-side scripting language and is commonly used to create dynamic web pages. It helps in changing web page contents dynamically, validating forms, etc. In this lab, we will create an HTML form that uses JavaScript to validate the user input.
	Javascript - Browser Console	The purpose of this lab is to practice using JavaScript in the browser console, to reinforce your understanding of certain concepts.

	Simple Interest Calculator	Simple Interest Calculator, to enhance your learning experience. Modify the HTML file as per the requirements. Modify the CSS file as per the requirements. Modify the JavaScript file as per the requirements. Verify that the webpage is working properly.
	Single Page Portfolio Website	In this project you will be creating a single page Portfolio website using HTML5, CSS & JavaScript having the following sections: About Me Skills Projects Recommendations
	Publish your Portfolio to Github Pages	In this lab, you will publish the portfolio page to Github Pages. GitHub Pages is a static site hosting service that takes HTML, CSS, and JavaScript files straight from a repository on GitHub, optionally runs the files through a build process, and publishes a website.
Getting Started with Git and GitHub	GitHub Sign Up and Create Repo	In this lab, you will get started with GitHub by creating a GitHub account and creating a new repository. You will then add a file to the repository using the GitHub web interface.
	Getting Started with Branches using Git Commands	You would typically use Git commands from your own desktop/laptop. However, so you can get started using the commands quickly without having to download or install anything, we are providing an IDE with a Terminal on the Cloud. Simply click the "Launch App" below to launch the Skills Network Cloud IDE and in the new browser tab that launches, follow the instructions to practice the Git commands. After completing this lab you will be able to use git commands to start working with creating and managing your code branches

	Cloning and Forking GitHub Projects	After completing this lab you will be able to use git commands to manage upstream repositories: fork existing repository using the UI clone forked repository in the lab environment create a new branch make changes locally add and commit to local branch push changes to your forked repository create a pull request to the upstream repository
	Practice Project: Overview	Assignment involves harnessing GitHub's collaborative and version control features to oversee the development and deployment of the shipping calculator web application. This initiative aims to lead to an optimized supply chain management process.
	Practice Project: Part 1 - GitHub UI	In this lab you will start your journey with the GitHub implemention for your organisation by creating a repository in your GitHub account and then initialising the repository with a README.md file and a License. Following this, you'll update the README file to include a Code of Conduct and Contribution Guidelines markdown files. After making these changes, you'll commit the files to your repository.
	Practice Project: Part 2 - Git CLI	All the developers have contributed to the repository, and their changes have been accepted and merged into a new global repository. Now, you have been asked to edit some of the code and also add few more files. For this, you will fork this repository, make the necessary edits, add files using Git CLI in the provided lab environment, and open a pull request.

	Final Project: Part 1 - GitHub UI	After completing this lab, you would have demonstrated that you can: Create a new repository in your GitHub account. Select the appropriate license for your project. Create a README.md file that explains the purpose of the project. Create a Code of Conduct markdown that explains how you want the community to behave and interact with each other. Create a Contribution Guidelines markdown that tells the community how to contribute. Commit the new files to the repository.
Developing Front-End Apps with React	Lab: Content Rating Application to Like or Dislike Content	In this lab, you will create a React component called 'ContentRating' where the component will let users rate material by clicking 'like' or 'dislike' buttons. When the component is rendered for the first time, both the like and dislike counts are set to zero. You will create a method that changes the state to add one to the number of likes when a user selects the 'like' button. In the same way, clicking the 'dislike' button, the method will add one to the number of dislikes. This action lets users rate the content in an interactive way and provides useful input to content creators.
	Lab: Function Component Event Planner Landing Page	In this lab, you are going to create a landing page for an organization that plans personal and corporate events. You will create the EventPlanner component that consists of various sections, such as the description of the application, event categories, features, testimonials, contact form, and footer. Each section describes various aspects of the Event Planner app, such as the types of events that can be planned, the app's main features, customer reviews, and how to get in touch with the service provider.
	Lab: Fetch Data Using useFetch	In this lab, you will learn how to fetch data in React using a custom hook (UseFetch) and a corresponding component (FetchData). By encapsulating data-fetching logic into reusable hooks, you can improve code maintainability. You will master React's useState and useEffect hooks for managing state and asynchronous operations efficiently. Additionally, you learn to dynamically render fetched data on the UI with JSX, enabling the creation of dynamic and interactive components.

	Lab: Create Feedback Form for Survey	In this lab, you will create a feedback form using React functional components and manage user details using the useState hook. You will implement event handlers to manage form input changes, validate user inputs, and handle form submissions. Additionally, you will create a confirmation dialog using the confirm method to confirm user details before final submission. Upon successful submission, you will reset the form fields and display a thank you message to the user. This lab will give you practical experience building interactive forms and handling user inputs in React applications.
	Practice Project: Conference Event Planner	The application's requirements include allowing users to select and price the rooms in the conference center, add-on selections, like microphones and projectors, and meals for a given number of guests.
Developing Back-End Apps with Node js and Express	First Server with ServerSide Java Script	Objective for Exercise: Use the terminal to git clone and get Node.JS server code Create a web server using Server side Java script Run the server Access the server from the client and get a response from server
	Promises and Call Back	After completing this lab, you will be able to: Describe Promise callbacks Create a Node.js application with promises
	Async Callback Programing	By the end of this lab, you will be able to understand asynchronous callbacks and be able to write the code in a Node.js application.

	CRUD Operations with Node.js and Express	In this lab you will learn how to create a Friend's list using Express server. Your application should allow you to add a friend with the following details: First name, Last name, Email and Date of birth. You will also be providing the application the ability to retrieve details, change details and delete the details. You will be creating an application with API endpoints to perform Create, Retrieve, Update and Delete operations on the above data using an Express server. You will also learn to provide authenticated access to the endpoints. You will use cURL and Postman to test the implemented endpoints.
	Express Server	Objective for Exercise: Create express server and run it Work on Middlewares with Express server Use middleware and JWT for authentication Render a static HTML page through express server
	Friends List Application Using Express Server with JWT	In the CRUD lab you performed CRUD operations on transient data by creating API endpoints with an Express Server. In this lab, you will restrict these operations to authenticated users using JWT and session authentication. In this lab, the friends object will be a JSON/dictionary with email as the key and friends object as the value. The friends object is a dictionary with firstName,lastName, DOB mapped to their respective values. You will thus be using “body” from the HTTP request instead of “query” and “params”. Only authenticated users will be able to perform all the CRUD operations. We will be testing the output of the endpoints on Postman.
	Book Review Application	In this final project, we will build a server-side online book review application and integrate it with a secure REST API server which will use authentication at session level using JWT. You will then test your application using Promises callbacks or Async-Await functions.

PCAC2204 CLOUD COMPUTING FOUNDATION LAB. (0-0-3)

Module Title	Lab Name	Details
Introduction to Cloud Computing	Obtain IBM Cloud Feature Code and Activate Trial Account	To facilitate hands-on skills development and enable you to complete some of the labs in this course, we are providing you with a special Feature Code that will enable you to create an IBM Cloud Trial Account. This feature code will enable you to either create an IBM Cloud trial account without a credit card. Please check the box and click on the "Open tool" button below to get a unique Feature Code to activate your IBM Cloud trial account.
	Deploy an Application on Cloud	In this hands-on lab you will build a docker container image, upload it to IBM Cloud Container Registry, and deploy an application on cloud using a serverless technology called IBM Code Engine! No programming knowledge required. Simply follow the step-by-step instructions and copy and execute the commands provided in the cloud-based lab environment from within your web-browser.
Cloud Computing Concepts, Part 1	Programming Assignment	By completing this programming assignment, you will be able to: Design, implement, and test your distributed failure detection protocol. Design, implement, and test your distributed membership protocol.
	Update notes for Gossip Protocol submission	We've made some system compatibility updates to the Gossip Protocol submission scripts. Before you try to submit your Gossip Protocol assignment, please download the attached file and extract it. Inside, you will find these files: mp1-regen-data run.sh submit.py

PCAC2205 PROGRAMMING INTERNET OF THINGS LAB. (0-0-3)

Module Title	Lab Name	Details
Introduction to the Internet of Things and Embedded Systems	Wireshark	Download and install Wireshark on a computer. Start Wireshark and start a packet capture. Open a browser on your computer and go to any webpage. Stop the Wireshark packet capture and examine it to find the first TCP packet which is recorded and whose source is your computer. Find the port number, P, used on your computer. Use the Wireshark filter to show only the messages to/from this port (tcp.port == P). Select the message and make sure that the Packet Details Pane is visible and showing the following information: 1) the Internet Protocol header with the Src and Dst IP addresses, and 2) the Transmission Control Protocol header with the Src Port number and Dst Port numbers. Print the screen image and submit it for grading. Also, submit one page which states your machine's IP address at the time of capture, and the port number that you found being used for the TCP connection on your computer.
The Arduino Platform and C Programming	Install Arduino IDE	install the Arduino IDE on your computer, compile the "Blink" example, upload the example to the board, and ensure that the LED blinks.
	Program to compute Fibonacci sequence	Write a program in C that computes and prints out the first six digits in the Fibonacci sequence.
	Arduino Blink	Write a program that causes the built-in LED connected to pin 13 on the Arduino to blink, alternating between fast blinks and slow blinks. The LED should blink 5 times quickly and then it should blink 5 more times slowly. The LED should continue to blink in this alternating fashion for as long as the Arduino receives power.
	Serial on Arduino	Write a program that allows the user to control the LED connected to pin 13 of the Arduino. When the program is started, the LED should be off. The user should open the serial monitor to communicate with the Arduino. If the user sends the character '1' through the serial monitor then the LED should turn on. If the user sends the character '0' through the serial monitor then the LED should turn off.

The Raspberry Pi Platform and Python Programming for the Raspberry Pi	Raspberry Pi setup	Setup your Raspberry Pi by usingNOOBSto install Raspbian on the micro SD card. Boot your Raspberry Pi to the desktop and take a picture (with a regular camera/phone) of the desktop. Submit the picture as evidence that you completed the task.
		Boot the Raspberry Pi and install the “scrot” program to take screen shots. You can install it by typing “sudo apt-get install scrot” in a terminal window. Use the scrot program to take a screenshot of your Raspberry Pi.
	Use Python shell	Write a Python program that prompts the user to input 3 numbers, one at a time. The Python program should put the numbers in a list, sort the list, and print the sorted list.
	Circuit using Raspberry Pi	Build a circuit using your Raspberry Pi that causes an LED to blink when a push button is NOT pressed. However, the LED should stay on continually when the push button IS pressed. Your video should show the LED blinking when the push button is not pressed, and it should show that the LED is constantly on while the button is pressed.

PCAC2206 ROBOTICS : MOTION PLANNING LAB. (0-0-3)

Module Title	Lab Name	Details
Robotics: Aerial Robotics	1-D Quadrotor Control	Get started with a 1D quadrotor simulator and and implement a controller for it. We will be testing your controller with two cases. In the first test case, the quadrotor simply needs to stabilize at a height of 0. The second test case gives the quadrotor a step input of 1 meter; that is, your quadrotor will be asked to rise to a height of 1 meter.
	2-D Quadrotor Control	Design and implement a controller for a planar (2D) quadrotor. We will be testing your controller with two trajectories, the first one is a horizontal line trajectory and the second one is a more complex sine wave shaped trajectory.
	3-D Quadrotor Control	Extend the controller you developed earlier for a full 3D quadrotor. In addition, you will write a trajectory generator which outputs time parameterized trajectories which allow a quadrotor to fly through a given set of waypoints.
Robotics: Computational Motion Planning	MATLAB Practice	MATLAB Grader: write a function to create a rotation matrix to show how to complete programming assignments.
	Dijkstra Problem	Write Matlab code to implement the Dijkstra planning algorithm that works on 2D grid like environments.
	A Star Algorithm	Write Matlab code to implement the A-star planning algorithm that works on 2D grid-like environments.
	Configuration space	Write a program to help guide a two link robotic arm from one configuration to another while avoiding obstacles in the workspace.
	Random Sampling Approaches	Write a program to help guide the six link robot shown in the figure from one configuration to another while avoiding the objects in the workspace.
	Gradient based planner	Develop a code to guide a robot from one location to another in a 2-dimensional configuration space using artificial potential fields.

PCAC2207 IT FUNDAMENTALS FOR CYBERSECURITY - I LAB. (0-0-3)

Module Title	Lab Name	Details
Introduction to Cybersecurity Tools & Cyberattacks	Hands-on Lab: Malware Scan Using Rkhunter	After completing this lab, you will be able to: Install rkhunter on a Linux system Run a malware scan to check for rootkits and malware
Operating Systems: Overview, Administration, and Security	Using File Explorer to Manage Files and Folders	n this hand-on lab, you will: Employ Microsoft Windows File Explorer for file and folder navigation Construct a new folder called “Office files” Use LibreOffice to create a new document and to add text to that document Use the “Office files” folder structure as the LibreOffice document file location Identify the stored path of the new Project 1 document
	Exploring Microsoft Windows Server Features	After completing this lab, you will be able to: Open and work with the Server Manager management console Access remote, multi-server management
	Using Microsoft Windows Command Prompt Tools for Administration	In this hands-on lab, you will: Obtain system information using the command line. Manage directories using the command line. Perform system maintenance using the command line.
	Creating Windows Users and Groups	In this lab you will familiarize yourself with Windows User Account creation on the Windows Server using the Server Manager.
	Windows Defender Firewall	In this lab, you will learn to: Configure the Windows Defender Firewall using the basic user interface.

	Creating a Firewall Rule in Microsoft Windows Defender	In this hands-on lab, you will: Use Windows Defender Firewall with Advanced Security to edit an existing firewall rule. Enforce the following rules: – Allow the connection for Key Management Service on the Domain and Private network. – Allow the connection for Key Management Service on the Public network.
	Getting Started with Linux Terminal	In this lab, you will Interact with the Linux Terminal, navigate directories on a Linux filesystem and explore their contents, install and update packages, create and edit files using `nano`, execute shell commands and applications from the terminal.
	Common Linux/Unix Commands	In this lab, you will be introduced to the use basic Unix commands related to the following categories: General purpose commands. Directory management commands. File management commands. Access control commands. Text processing commands. Networking commands.
	Linux User Management	In this lab you will learn how to do create users, create groups and add user to groups within Kali Linux. Kali Linux is a well-known, open-source Linux operating system built on the Debian platform. It's tailored for professionals and learners in the field of cybersecurity and is purposefully crafted for tasks like penetration testing and ethical hacking.
	Updating Linux	After completing this lab, you will know how to: Update package lists Perform Linux system upgrade
	Getting Started with Linux Terminal	In this lab, you will Interact with the Linux Terminal, navigate directories on a Linux filesystem and explore their contents, install and update packages, create and edit files using `nano`, execute shell commands and applications from the terminal.
	Common Linux/Unix Commands	In this lab you will practice working with commonly used Linux commands for directory management, file management, access control, text processing, and networking .

	Linux User Management	In this lab, you will practice working with commonly used Linux commands for user management.
	Updating Linux	In this lab, you will perform important updates needed to secure Linux systems.
	Introduction to Containers, Docker and IBM Cloud Container Registry	This lab will demonstrate how to pull an image from Docker Hub and run an image as a container using docker, and build an image using a Dockerfile and push an image to IBM Cloud Container Registry.
	Final Project Part 1: Windows	After completing this project, you will have demonstrated your ability to Create a new user and a new group using Windows Server Manager Add a user to a group using Windows Server Manager Check for updates to virus and threat protection using a Windows operating system Run a "Quick Scan" to verify up-to-date virus and threat protection on a Windows operating system Create an inbound rule that controls connections on a TCP port using Windows Defender Firewall
	Final Project Part 2: Linux Tasks	After completing this project, you will have demonstrated your ability to Create a new user on a Linux operating system Manage files and directories on a Linux operating system Apply system updates on a Linux operating system