

ಸ್ಥಾಪನೆ: ೨೦೧೭

Centenary Celebrated Sharnbasveshwar Vidya Vardhaka Sangha's

Estd. : 2017



ಶರಣಬಸವ
SHARNBASVA
ಕಲಬುರಗಿ - ೫೮೫ ೧೦೩



ವಿಶ್ವವಿದ್ಯಾಲಯ
UNIVERSITY
KALABURAGI-585 103



A State Private University approved by Govt. of Karnataka vide Notification No. ED 144 URC 2016 dated 29-07-2017
Recognised by UGC under Section 2f vide No. F.8-29/2017 (CPP-I/PU), dated 20-12-2017 & AICTE, CoA, PCI New Delhi

CURRICULUM

Outcome Based Education

2021 - 2022

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

III & IV SEMESTER B. Tech.

**FACULTY OF ENGINEERING ANF TECHNOLOGY
(EXCLUSIVELY FOR WOMEN)**

SHARNBASVA UNIVERSITY, KALABURAGI

About Sharnbasva University

Sharnbasva University established in the year 2017, is the first private multi-disciplinary university in the entire North Karnataka providing avenues for students to pursue higher education in almost all fields of education such as Engineering, Management, Science, commerce, Architecture, Pharmaceutical Sciences, Humanities, and Social Sciences. Providing quality education at an affordable cost has been the hallmark of the university. The university has set itself on the path to one of the emerging centres of excellence and a world-class University, which is the dream of Poojya Dr. Sharnbaswappa Appa, the visionary chancellor of the youngest private university in the entire state.

The Sharnbasva University is backed by the rich legacy of more than 100 years of providing quality education to the deprived sections of the people in the educationally backward Kalaburagi and neighbouring districts. The Sharnabasaveshwar Vidya Vardak Sangha, under whose aegis the Sharnbasva University is functioning, has the honor of starting the first school exclusively for girls way back in 1934 opening doors of learning to the girls. The legacy of providing optimum importance to girl education has continued to this day and the Sangha was the first to start an exclusive engineering college for women in the entire North Karnataka and the second in the entire Karnataka. This College is now part of Sharnbasva University. The university houses five Center of Excellence (CoE) in Artificial Intelligence, Machine Learning, the Internet of Things, Drone Technology, 5G and 6G Technology Lab.

The institution offers:

- 4-year undergraduate program - Bachelor of Technology (in Artificial Intelligence & Data Science, Civil, Computer Science, Computer System Design, Energy Engineering, Electronics & Communication, Electrical & Electronics, Mechanical).
- 4-year undergraduate program (Exclusively for Women) - Bachelor of Technology (in Artificial Intelligence & Machine Learning, Civil, Computer Science, Electronics & Communication, Electrical & Electronics, Mechanical).
- 2-Year Post Graduate Programs - Master of Technology (Structural Engineering, VLSI Design & Embedded Systems, Machine Design, Computer Science and Engineering, Computer Network Engineering, Artificial Intelligence and Data Science, Digital Electronics) and Ph. D Programs
- 2-Year Post Graduate Programs - Master of Technology (Exclusively for Women) (Computer Science and Engineering, Digital Communication and Networking) and Ph. D Programs
- 5 Years Under program – Bachelor of Architecture (B. Arch.)
- 3-Year undergraduate programs – Bachelor of Computer Applications (BCA), – Co-Education and Exclusively for Women, Bachelor of Business Administration (BBA) BBA (HR, Finance and Marketing)– Co-Education and Exclusively for Women, BBA (Logistics), BBA (Tourism & Travel Mgmt.), BBA (Aviation and Air cargo) for Exclusively for Women.

- 2-Year Post Graduate programs - Master of Computer Application (MCA) and Ph. D Programs.
- 2-Year Post Graduate programs - Master of Business Administration (MBA) – Co-Education and Exclusively for Women, MBA (Tourism), MBA (Hospital Administration). Master of Commerce and Ph. D programs.
- 2-Year Post Graduate Programs – Master of Arts (MA) – MA (Kannada, English), MA (Fine Arts), MA (Journalism and Mass Comm.), MA (Music), MA (Visual Arts) and Ph. D Programs.
- 2 Year Post Graduate Programs – Master of Science (M. Sc.) – M. Sc. (Physics), M. Sc. (Mathematics), M. Sc. (Botany), M. Sc. (Zoology) and Ph. D Programs.

University Vision:

- Enhancing the horizon of world knowledge by promoting international understanding through imparting quality education and creating value added skill based human resource.
- We aspire to become top ranking National and International Centre of Excellence producing high caliber leaders and captains of Industry.
- Inculcating the spirit of "Vasudaiva Kudumbakam" (The world is one family).
- "Sakala Jeevathmarige Lesane Bayasuva" (Wishing the worldly good and betterment of all the living beings) is our moto.

University Mission

- Achieving academic excellence through innovatively designed, research intensive, industry-oriented education.
- Imbibing the culture of independent thinking, independent writing, independent speaking and independent living among the students in all the activities.
- Foster the spirit of National development and develop global competencies. Nurture creativity and encourage entrepreneurship and provide an education with rigor and relevance.
- Provide academically excellent, time efficient, and cost-effective higher education. Provide an education which enhances the ability of students to learn throughout the life.
- Ensure freedom of thought and expression and a campus without discrimination. Encourage the spirit of questioning and ensure close inter-relationship between Teaching, Scholarship and Research.
- Develop and deliver distinctive and value driven academic programme that are flexible and responsible to Local, National and International needs.
- Cultivate academic, business and community driven partnership that positions the University as a leading choice for adult learners.
- To work effectively with other institutions and organisations, where such partnerships can lead to understanding research and teaching.

About Faculty of Engineering and Technology (Exclusively for Women):

The Faculty of Engineering and Technology (Exclusively for Women) (FETW), formerly known as Godutai Engineering College for Women, was established in 2017. It is the only women's engineering college in the entire northern backward part of Karnataka state, meeting the rapidly growing need for women professionals in technology. Nurturing young minds in an innovative and progressive learning environment, we are committed to promoting technical education as a catalyst for the country's and society's growth and development, a dream of our Chancellor and President, Poojya Dr. Sharnbaswappa Appa. With world-class infrastructure and experienced faculty, the Faculty of Engineering and Technology is the preferred destination for technocrats shaping the future.

FETW offers the following B.Tech. (Undergraduate) programs:

- Artificial Intelligence and Machine Learning (AI & ML) – 60 seats
- Civil Engineering (CV) – 60 seats
- Computer Science and Engineering (CSE) – 150 seats
- Electrical and Electronics Engineering (EEE) – 60 seats
- Electronics and Communication Engineering (ECE) – 120 seats

M. Tech. (Postgraduate) programs:

- Computer Science and Engineering (CSE) – 18 seats
- Digital Communication and Networking (DCN) – 18 seats

Research Programs (Ph.D.):

- Computer Science and Engineering
- Electronics and Communication Engineering
- Electrical and Electronics Engineering
- Basic Sciences (Physics, Chemistry, Mathematics)

Vision of the Institute

We aspire to become global model for women professional through quality education and ethical values in the field of engineering and technology.

Mission of the Institute

- **To inspire a research culture, encourage entrepreneurial efforts and empower globally to be great leaders.**
- **To create technical women's power to meet the current and future demand of the industry.**
- **To develop women professionals with good academic knowledge, technical skills, strong ethics and above all good human being.**

About the Department:

The Department of B. Tech. in Artificial Intelligence and Machine Learning (AI and ML) at the Faculty of Engineering and Technology (Exclusively for Women) was established in the year 2021 with an intake of 60 students. It is a four-year full-time undergraduate program recognized by the All-India Council for Technical Education (AICTE). Artificial Intelligence and Machine Learning aim to develop a strong foundation by using principles and technologies that consist of many facets of Artificial Intelligence, including logic, knowledge representation, probabilistic models, and machine learning.

To nourish the young minds, the department has a team of dedicated and experienced faculty with good infrastructure, curriculum, comprehensive hands-on experience, and conducting research in the field of Artificial Intelligence & Machine Learning.

The department has well-equipped laboratories with higher configuration systems along with a smartboard available in the laboratory to provide the best to the students. The department has a wide technical spectrum through the centre of excellence, special labs, industrial tie-ups, collaborations, MOUs, and online forums.

Vision

To train women professionals to create sustainable, intelligent solutions that work with natural intelligence to empower future generations digitally while protecting social ethics.

Mission

1. To provide female students with the necessary foundational skills to pursue a successful career in the field of computer science and engineering, with a focus on artificial intelligence and machine learning.
2. To Improve academic outcomes through location-based education and relationships with established research laboratories and enterprises.
3. To Inspire young women to pursue entrepreneurial endeavours and help them grow into influential leaders.

Program Educational Outcomes (PEOs)

After a few years of graduation with a B. Tech. in Artificial Intelligence and Machine Learning, the girl graduates will be able:

- PEO-1: To analyse the problems by applying the principles of computer science, mathematics, and scientific investigation to build intelligent machines, software, or applications using the latest technologies to solve problems of the real world.
- PEO-2: To start an enterprise to improve the economy of the country by providing support to the customer for a lifelong learning attitude.

- PEO-3: To Pursue higher studies with research potential in the field of Artificial intelligence, Engineering in allied areas and continue to learn by participating in conferences, and seminars, etc.

Program Outcomes (POs)

On successful completion of the program, the graduates of the B. Tech. in Artificial Intelligence and Machine Learning program will be able to:

- **PO 1: Engineering knowledge:** Apply the knowledge of mathematics, science, and engineering fundamentals for the solution of complex problems in Artificial Intelligence and Machine Learning.
- **PO 2: Problem analysis:** Identify, formulate, research literature, and analyze engineering problems to arrive at substantiated conclusions using the first principles of mathematics, nature, and engineering sciences.
- **PO 3: Design/development of solution:** Design solutions for complex engineering problems and design system components, and processes to meet the specifications with consideration for public health and safety, and cultural, societal, and environmental considerations.
- **PO 4: Conduct Investigation of a complex problem:** Use research-based knowledge including design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO 5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- **PO 6: Engineering and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **PO 7: Environment and sustainability:** Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- **PO 8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
- **PO 9: Individual and Teamwork:** Function effectively as an individual, and as a member or leader in teams, and multidisciplinary settings.
- **PO 10: Communication:** Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective report documentation. Make effective presentations and give and receive clear instructions.
- **PO 11: Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to

one's work, as a member and leader in a team. Manage projects in multidisciplinary environments.

- **PO 12: Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program-Specific Outcomes (PSOs)

Upon successful completion of the B. Tech. in Artificial Intelligence and Machine Learning (AI and ML) program, graduates will possess the following abilities:

PSO 1: Understand, analyze, and develop vital skills in areas related to computer science and artificial intelligence, with a focus on underlying statistical and computational principles. Apply this knowledge to solve practical problems.

PSO 2: Demonstrate the ability to implement AI and ML techniques, including search algorithms, neural networks, machine learning, and data analytics, for solving problems and designing novel algorithms. Develop skills essential for a successful career and entrepreneurship.

PSO 3: Utilize modern tools and techniques in computer science, artificial intelligence, and data science.

Semester wise Credit Breakdown for B. Tech. Degree Curriculum Batch 2021-24

	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eight	Total Credits
Basic Sciences (BSC)	7	7	3	3	-	-	-	-	20
Engineering Sciences (ESC)	10	10	-	-	-	-	-	-	20
Humanities, Social Sciences and Management (HSMC)	1	1	1	1	-	4	1	-	9
Ability Enhancement Course (AEC)	1	1	1	1	1	1	-	-	6
Professional Core Courses (PCC)	-	-	15	15	12	5	12	-	59
Professional Elective Courses (PEC)	-	-	-	-	3	6	3	-	12
Institutional Open Elective Courses (IOE)	-	-	-	-	4	4	4	-	12
Internship (INT)	-	-	-	-	-	-	-	7	7
Mini Project / Project Work (PW)	1	1	1	1	1	1	1	8	15
Total Credits	20	20	21	21	21	21	21	15	160

Sharnbasva University, Kalaburagi
Scheme of Teaching and Examination 2021-22
Outcome Based Education (OBE) and Choice Based Credit System (CBCS), (Effective from the academic year 2021-22)

Programme: B.Tech.: Artificial Intelligence and Machine Learning

III SEMESTER

Sl. No.	Course Code		Course Title	Teaching Department	Teaching Hours/week			Examination				Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P					
1	BS	21MAT31	Engineering Mathematics III	Mathematics	3			3	50	50	100	03
2	PCC	21CS32	Logic Design and Computer Architecture	CSE/ISE	3	1		3	50	50	100	04
3	PCC	21CS33	Data Structures and Its Applications Using C	CSE/ISE	2	1		3	50	50	100	03
4	PCC	21AM34	Operating System	CSE/ISE	2	1		3	50	50	100	02
5	PCC	21AM35	Python Programming	CSE/ISE	1	1		3	50	50	100	03
6	PCC	21CSL36	Data Structures and Its Applications Using C Lab	CSE/ISE			2	3	50	50	100	01
7	PCC	21AML37	Operating System Lab	CSE/ISE			2	3	50	50	100	01
8	PCC	21AML38	Python Programming Lab	CSE/ISE			2	3	50	50	100	01
9	PW	21PRJ39	Project-III	CSE/ISE			2	3	50	50	100	01
10	HSS	21KANKK310 /21KANAK310	Kannada Kali-III/ AydaKategalu	Humanities	1			2	50	50	100	01
11	AEC	21AAM311X	Ability Enhancement Course-III				2	3	50	50	100	01
Total					12	4	10	32	550	550	1100	21

Note: BS-Basic Science, PCC- Programme Core Course, PW-Project Work, AEC- Ability Enhancement Course, HSS-Humanity and Social Science, NCMC- Non-Credit Mandatory Course

21KANKK310 Kannada Kali-III is for non Kannada speaking, reading and writing students and 21KANAK310 AydaKategalu is for the students who speak, read and write Kannada.

Project (PRJ): Based on the ability /abilities of the student/s and recommendations of the mentor, a single discipline or multidisciplinary mini project assigned to individual students or to a group having not more than 4 students.												
Ability Enhancement Course-3												
Course code under 21AEC311X				Course Title								
21AAM3111				Unix and Shell Programming								
21AAM3112				Scilab								
Courses prescribed to lateral entry Diploma holders admitted to III semester of Engineering programs												
12	NCMC	21MATDIP31	Additional Mathematics- I	Mathematics	3	1	-	3	00	100	100	00
Non-Credit Mandatory Courses (NCMC) Additional Mathematics-I and II prescribed for III and IV semesters respectively, to the lateral entry Diploma holders admitted to III semester of B. Tech. programs, shall attend the classes during the respective semesters to complete all the formalities of the course and appear for the university examination. In case any student fails to register for the said course/fails to secure the minimum 50% of the prescribed CIE marks, he/she shall be deemed to have secured F grade. In such a case, the students have to fulfill the requirements during subsequent semester/s to appear for SEE.												
These courses shall not be mandatory for vertical progression, but completion of the courses shall be mandatory for the award of degree.												
Courses prescribed to lateral entry B.Sc. degree holders admitted to III semester of Engineering programs												
Lateral entry students from B.Sc. stream shall clear the non-credit courses Computer Aided Engineering Drawing, Elements of Civil Engineering of First Year Engineering Programme. These Courses shall not be considered for vertical progression, but completion of the courses shall be mandatory for the award of degree.												
AICTE Activity Points to be earned by students admitted to B.Tech. programme (For more details refer to Chapter 6, AICTE Activity Point Programme, Model Internship Guidelines):												
Over and above the academic grades, every regular student admitted to the 4 years Degree programme and every student entering 4 years Degree programme through lateral entry, shall earn 100 and 75 Activity points respectively for the award of a degree through AICTE Activity Point Programme. Students transferred from other universities of fifth semester are required to earn 50 activity points from the year of entry to Sharnbasva University. The Activity Points earned shall be reflected on the students eighth semester Grade card.												
The activities can be spread over the years, anytime during the semester weekends holidays, as per the liking and convenience of the student from the year of entry to the programme. However, minimum hours requirement should be fulfilled. Activity Points (non-credit) have no effect on SGPA/CGPA and shall not be considered for vertical progression.												

Sharnbasva University, Kalaburagi
Scheme of Teaching and Examination 2021-22
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
(Effective from the academic year 2021-22)

Programme: B.Tech.: Artificial Intelligence and Machine Learning

IV SEMESTER

Sl. No.	Course Code		Course Title	Teaching Department	Teaching Hours/week			Examination				Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P					
1	BS	21MAT41	Engineering Mathematics IV	Mathematics	3			3	50	50	100	3
2	PCC	21AM42	Microcontroller And Embedded Systems	CSE/ISE	3			3	50	50	100	3
3	PCC	21AM43	Design and Analysis of Algorithms	CSE/ISE	2	2		3	50	50	100	3
4	PCC	21AM44	Introduction to Artificial Intelligence	CSE/ISE	2	2		3	50	50	100	3
5	PCC	21AM45	Foundations of Data Science	CSE/ISE	2	2		3	50	50	100	3
6	PCC	21AML46	Design And Analysis of Algorithms Lab	CSE/ISE			2	3	50	50	100	1
7	PCC	21AML47	Artificial Intelligence Lab	CSE/ISE			2	3	50	50	100	1
8	PCC	21AML48	Data Science Lab	CSE/ISE			2	3	50	50	100	1
9	PW	21PRJ49	Project-IV	CSE/ISE			2	3	50	50	100	1
10	HSS	21KANKK41/ 21 KANMD410	Kannada Kali-IV/ Mahadasohigalu	Humanities	1			2	50	50	100	1
11	AEC	21AAM411X	Ability Enhancement Course-IV				2	3	50	50	100	01
Total					12	4	10	32	550	550	1100	21

12	NCMC	21MATDIP41	Additional Mathematics - II*	Mathematics	3	1	-	3	00	100	100	00
Note: BS-Basic Science, PCC- Programme Core Course, PW-Project Work, AEC- Ability Enhancement Course, HSS-Humanity and Social Science, NCMC-Non-Credit Mandatory Course												
21KANKK410 Kannada Kali-IV is for non-Kannada speaking, reading and writing students and 21KANMD410 Mahadasohigalu is for the students who speak, read and write Kannada.												
Project(PR): Based on the ability /abilities of the student/s and recommendations of the mentor, a single discipline or multidisciplinary mini project can be assigned to an individual student or to a group having not more than 4 students.												
Courses prescribed to lateral entry Diploma holders admitted to III semester of Engineering programs												
Course code under 21AEC411X						Course Title						
21AAM4111						Scripting Languages						
21AAM4112						Data Analytics with Excel						
<p>* Lateral Entry Students: Non-Credit Mandatory Courses (NCMC) Additional Mathematics-I and II prescribed for III and IV semesters respectively, to the lateral entry Diploma holders admitted to III semester of B. Tech. programs, shall attend the classes during the respective semesters to complete all the formalities of the course and appear for the university examination. In case any student fails to register for the said course/fails to secure the minimum 50% of the prescribed CIE marks, he/she shall be deemed to have secured an F grade. In such a case, the students have to fulfill the requirements during subsequent semester/s to appear for SEE. These courses shall not be mandatory for vertical progression, but completion of the courses shall be mandatory for the award of degree.</p>												
Courses prescribed to lateral entry B.Sc. degree holders admitted to III semester of Engineering programs.												
<p>Lateral entry students from B.Sc. stream, shall clear the non-credit courses Computer Aided Engineering Drawing, Elements of Civil Engineering of First Year Engineering Programme. These Courses shall not be considered for vertical progression, but completion of the courses shall be mandatory for the award of a degree.</p> <p>AICTE Activity Points to be earned by students admitted to B.Tech. programme (For more details refer to Chapter 6, AICTE Activity Point Programme, Model Internship Guidelines): Over and above the academic grades, every regular student admitted to the 4 years Degree programme and every student entering 4 years Degree programme through lateral entry, shall earn 100 and 75 Activity points respectively for the award of a degree through AICTE Activity Point</p>												

Programme. Students transferring from other universities to the fifth semester are required to earn 50 activity points from the year of entry to Sharnbasva University. The Activity Points earned shall be reflected on students eighth semester Grade card. The activities can be spread over the years, anytime during the semester weekends holidays, as per the liking and convenience of the student from the year of entry to the programme. However, the minimum hours requirement should be fulfilled. Activity Point non-credit) has no effect on SGPA/CGPA and shall not be considered for vertical progression.

ENGINEERING MATHEMATICS - III

(Effective from the academic year 2021-2022)

SEMESTER - III

Course Code	21MAT31	CIE Marks	50
Number of Lecture Hours/Week (L: T:P)	3:0:0	SEE Marks	50
Total Number of Lecture Hours	40	Exam Hours	3
CREDITS - 03			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> • Introduce most commonly used analytical and numerical methods in the different engineering fields.- • Learn Laplace transform and Z-transforms to solve ODE and PDE's. • Understanding the statistical methods, numerical methods. • Solve the problem related to Interpolation. • To discuss the random variable and associated probability distributions. • Understand the vector space and associated results. • Understand the basic concepts of set theory, relations, functions and mathematical logic. 			
Module I			8 Hours
<p>Laplace Transforms: Definition, Laplace transforms of Elementary functions, properties (without proof) periodic function, Unit step function, Unit impulse function.</p> <p>Inverse Laplace Transforms: Definition, Convolution Theorem (without proof) and Finding Inverse Laplace transform by convolution Theorem. Solution of Linear Differential equations using Laplace Transforms and Applications (5 Assignment Problem).</p> <p>Self Study : Solution of first order simultaneous differential equation</p>			
Module II			8 Hours
<p>Probability Distribution: Random variables (discrete and continuous) probability mass/density functions. Binomial distribution, Poisson distribution. Exponential and Normal distributions. Problems. (5 Assignment Problem).</p> <p>Self Study: Definition of probability, addition and multiplication rule, Bay's theorem.</p>			
Module III			
<p>Statistical Methods: Correlation-karl Pearson's co-efficient of correlation problems. Regression analysis lines of regression, Rank correlation (without proof)-problems.</p> <p>Curve Fitting: Curve fitting by the method of least square. Fitting of the curves of the form $y = ax + b$, $y = ax^2 + bx + c$ & $y = ae^{bx}$.</p> <p>Numerical Methods: Numerical solution of algebraic and transcendental equations by Regula-Falsi Method and Newton-Raphson method. (5 Assignment Problem).</p> <p>Self Study : Secant method, mean, mode, median, variance and standard deviation.</p>			
Module IV			
<p>Finite Difference: Forward and Backward differences, Newton's forward and backward interpolation formulae. Divided difference-Newton's divided difference formulae.</p>			

<p>Lagrange's-interpolation formula and inverse interpolation formula (all formula without proof) problems.</p> <p>r^{th} t^{th}</p> <p>Numerical Integration: Simpsons(1), (3) rules, Weddle's rule (without proof) problems. (5 Assignment Problem).</p> <p>Self Study : Numerical differentiation, Trapezoidal rule</p>	
Module V	8 hours
<p>Functions: Cartesian Products and Relations, Functions – into, many one One-to-One, Onto, Bijective Functions. The Pigeon-hole Principle, Function Composition and Inverse functions.</p> <p>Relations: Definition and different types of relations</p> <p>Introduction to logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference.</p> <p>Self Study : Different types of sets and operations on sets</p>	
<p>Course Outcomes: After studying this course, students will be able to:</p>	
<p>CO1: Apply the knowledge of Laplace transform from time domain to frequency domain. Knowing the property of Laplace transform and solving the problems on Signal and image processing which transforms differential equation into algebraic equation form and solving the problems also in inverse Laplace transform.</p> <p>CO2: Knowing the random variable both discrete and continuous and their probability distribution, Mass density function and solving the problems on various engineering problems.</p> <p>CO3: Apply the concept of correlation and regression lines for solving the problems and numerical techniques to solve engineering problems and fit a least squares curve to the given data.</p> <p>CO4: studying the forward and backward finite differences and solve the problems on Interpolation and Finding the numerical integration by different methods.</p> <p>CO5: Apply the knowledge of z-transforms in solving the difference equation arising in the continuous and discrete time signals and digital processing, Apply the knowledge of vector space in digital communication. Apply sampling distribution to solve engineering problems. Apply the operations like union and intersection on discrete structures such as sets, relations and functions and construct mathematical arguments using logical connectives.</p>	
<p>Question paper pattern:</p> <ul style="list-style-type: none"> • The question paper will have ten questions. • There will be 2 questions from each module. • Each question -will have questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. B.S. Grewal : Higher Engineering Mathematics, Khanna Publishers, 43rd Ed., 2015. 2. E. Kreyszig : Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed.(Reprint), 2016. 	
<p>Reference Books:</p> <ol style="list-style-type: none"> 1. C.Ray Wylie, Louis C.Barrett : "Advanced Engineering Mathematics", 6th Edition, 2. McGraw-Hill Book Co., New York, 1995. 	

2. James Stewart : "Calculus -Early Transcendentals", Cengage Learning India Private Ltd., 2017.
3. B.V.Ramana : "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
4. Srimanta Pal & Subobh C Bhunia:"Engineering Mathematics", Oxford University Press,3rd Reprint,2016.
5. Gupta C.B., Singh S.R. and Mukesh Kumar : "Engineering Mathematics for Semester I & II",Mc-Graw Hill Education (India) Pvt.Ltd., 2015.

Course outcomes

CO #	Course Outcome
CO1	Apply the knowledge of Laplace transform from time domain to frequency domain. Knowing the property of Laplace transform and solving the problems on Signal and image processing which transforms differential equation into algebraic equation form and solving the problems also in inverse Laplace transform
CO2	Knowing the random variable both discrete and continuous and their probability distribution, Mass density function and solving the problems on various engineering problems
CO3	Apply the concept of correlation and regression lines for solving the problems and numerical techniques to solve engineering problems and fit a least squares curve to the given data.
CO4	Studying the forward and backward finite differences and solve the problems on Interpolation and Finding the numerical integration by different methods.
CO5	Apply the knowledge o z-transforms in solving the difference equation arising in the continuous and discrete time signals and digital processing, Apply the knowledge of vector space in digital communication. Apply sampling distribution to solve engineering problems. Apply the operations like union and intersection on discrete structures such as sets, relations and functions and construct mathematical arguments using logical connectives.

BLOOM'S Level of The Course Outcomes

CO#	Bloom's Level					
	Remember (L1)	Understand (L2)	Apply (L3)	Analyze (L4)	Evaluate (L5)	Create (L6)
CO1	√	√	√			
CO2	√	√	√			
CO3	√	√	√			
CO4	√	√	√			
CO5	√	√	√			

Course Articulation Matrix

CO#/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO2
CO1	3	2	2		1				1			1			
CO2	3	2	2		1				1			1			
CO3	3	2	2		1				1			1			
CO4	3	2	2		1				1			1			
CO5	3	2	2		1				1			1			

Note: 1-Low, 2-Medium, 3-High

LOGIC DESIGN AND COMPUTER ARCHITECTURE (Effective from the academic year 2021-2022) SEMESTER - III			
Course Code	21CS32	CIE Marks	50
Number of Lecture Hours/Week (L:T:P)	4:0:0	SEE Marks	50
Total Number of Lecture Hours	50	Exam Hours	3
CREDITS - 04			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> • Describe, Illustrate and Analyze Combinational Logic circuits, Simplification of Algebraic equations using Karnaugh Maps and Quine McClusky Techniques. • Describe, Design, and Analyze Encoders, Digital Multiplexers, Master-Slave Flip-Flops, and Synchronous and Asynchronous Sequential. • How Computer System works & the Basic Principles, Instruction Level Architecture, and Instruction Execution. • The Current state of the art in Memory system design How I/O devices are accessed and its principles. • To provide knowledge on Instruction Level Parallelism and Understand Concepts of advanced pipelining techniques, Computer Arithmetic, and parallel Processing. 			
Module I			10 Hours
<p>Basic of Gates: Review of Basic Logic Gates, Positive and Negative Logic, Combinational Logic Circuits: Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, Encoders, Exclusive-or Gates, Parity Generators and Checkers.</p> <p>DEMONSTRATION:</p> <ol style="list-style-type: none"> 1. Design and implement Half adder, Full Adder, Half Subtractor, and Full Subtractor using basic gates. 2. Given a 4-Variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC. 			
Module II			10 Hours
<p>Flip-Flops: RS Flip-Flops Gated Flip-Flops, Edge-Triggered Flip-Flops, Flip-Flop Timings, JK Master-Slave Flip-Flop, and Various Representations of Flip-flops. Registers: Types of Registers, Application of Shift Registers. Counters: Asynchronous Counters, Synchronous Counters, Decade Counters, Counter Design as a Synthesis problem, and A Digital Clock. Design of Synchronous Sequential Circuit: Model Selection, State Transition Diagram, State Synthesis Table, Design Equations and Circuit Diagram, State Reduction Technique.</p> <p>DEMONSTRATION:</p> <ol style="list-style-type: none"> 1. Realize a J-K Master/Slave Flip-Flop using NAND gates and verify its truth table. 2. Design and implement code converter <ol style="list-style-type: none"> (1) Binary to Grey (2) Grey to Binary Code using Basic Gates 			
Module III			10Hours
Basic Structure of Computers: Functional Units, Basic Operational Concepts, Bus Structures, Performance. Machine instructions and Programs: Memory locations and addresses, Memory			

<p>operations, Instructions and instruction sequencing, Addressing modes, Assembly language, and basic input-output operations.</p> <p>Input-output Organization: Accessing I/O devices, Interrupts, and DMA.</p>	
Module IV	10Hours
<p>Memory Hierarchy: Introduction, Cache Performance, Six basic cache optimization, Virtual memory, and Memory hierarchy design: 10 advanced optimizations of cache performances.</p>	
Module V	10Hours
<p>Basic Processing Unit: Single bus organization, Multiple bus organization, Hardwired and micro-programmed design approaches.</p> <p>Pipelining: Introduction, Major hurdles of Pipelining, How is pipelining implemented? Instruction level Parallelism: Concepts and Challenges.</p>	
<p>Course Outcomes: After studying this course, students will be able to:</p>	
<p>CO1: Analyze and simplify combinational logic circuits using Boolean algebra, K-maps, and Quine–McCluskey methods.</p> <p>CO2: Design and evaluate sequential circuits using flip-flops, registers, and counters.</p> <p>CO3: Explain the structure and functional organization of computer systems and evaluate performance parameters.</p> <p>CO4: Analyze memory hierarchy, pipelining, instruction-level parallelism, and processor design concepts.</p>	
<p>Question paper pattern:</p> <ul style="list-style-type: none"> • The question paper will have ten questions. • There will be 2 questions from each module. • Each question -will have questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
<p>Textbooks:</p> <ol style="list-style-type: none"> 1) Carl Hamacher, Z. Vranesic&S.Zaky, “Computer Organization”,5th Edition, Tata McGraw Hill Publishing Company Ltd. New Delhi, 2002. 2) John L. Hennessy and David A. Patterson, Computer Architecture: A quantitative approach, 5th edition, Morgan Kaufmann Elsevier, 2013 3) Donald P Leach, Albert Paul Malvino& Goutham Saha: Digital Principles and Applications, 8th Edition, Tata Mcgraw Hill, 2015. 	
<p>Reference Books:</p> <ol style="list-style-type: none"> 1) Morris Mano, “Computer System Architecture”, PHI, 19862.William Stallings Computer Organization & Architecture, 7th Edition, PHI 2006. 2) Kai Hwang and Naresh Jotwani, Advanced Computer Architecture (SIE): Parallelism, Scalability, Programmability, McGraw Hill Education 3/e. 2015. 3) R D Sudhakar Samuel: Ilustrative approach to Logic Design, Sanguine-Pearson, 2010. 	

Course outcomes

CO #	Course Outcome
CO1	Analyze and simplify combinational logic circuits using Boolean algebra, K-maps, and Quine-McCluskey methods.
CO2	Design and evaluate sequential circuits using flip-flops, registers, and counters.
CO3	Explain the structure and functional organization of computer systems and evaluate performance parameters.
CO4	Analyze memory hierarchy, pipelining, instruction-level parallelism, and processor design concepts.

DATA STRUCTURES AND ITS APPLICATIONS USING C
(Effective from the academic year 2022-2023)

SEMESTER - III

Course Code	21CS33	CIE Marks	50
Number of Lecture Hours/ Week (L:T:P)	3:0:0	SEE Marks	50
Total Number of Lecture Hours	40	Exam Hours	3
CREDITS - 03			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> • To impart the basic concepts of data structures and algorithms. • To understand concepts about searching and sorting techniques • To understand basic concepts about stacks, queues, lists, trees, and graphs. • To enable them to write algorithms for solving problems with the help of fundamental data structures. 			
Module I			8 Hours
<p>Introduction: Data Structures, Classifications (Primitive & Non-Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory Dynamically allocated arrays.</p> <p>Array Operations: Traversing, inserting, deleting, searching, and sorting. Multidimensional Arrays, Polynomials, and Sparse Matrices.</p> <p>Strings: Basic Terminology, Storing, Operations, and Pattern Matching algorithms. Programming Examples.</p>			
Module II			8 Hours
<p>Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression.</p> <p>Recursion: Factorial, Fibonacci Sequence, Tower of Hanoi, Ackerman's function.</p> <p>Queues: Definition, Array Representation. Queue Operations, Circular Queues, Dequeues, Priority Queues. Multiple Stack and Queues, Programming Examples.</p>			
Module III			8 Hours
<p>Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage collection, Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists and header linked lists. Linked Stacks and Queues. Applications of Linked lists - Polynomials, Sparse matrix representation. Programming Examples.</p>			
Module IV			8 Hours
<p>Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations-copying binary tree, testing equality. Threaded binary trees, Binary Search Trees - Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Programming Examples.</p>			
Module V			8 Hours
<p>Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search.</p> <p>Sorting and Searching: Insertion Sort, Radix sort, selection sort. Hashing: Hash Table organizations, Hashing Functions.</p>			
Course Outcomes: After studying this course, students will be able to:			
CO1: Implement dynamic memory operations and pointer-based structures in C programs.			

CO2: Apply appropriate data structures and algorithms for organizing and processing data.

CO3: Use stacks, queues, linked lists, trees, and graphs for solving computational problems.

CO4: Implement and analyze data structures for problem-solving in a high-level programming environment.

CO5: Apply hashing techniques and collision-resolution strategies for efficient data retrieval.

•

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

- 1) Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
- 2) Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014

Reference Books:

- 1) Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning, 2014.
- 2) Reema Thareja, Data Structures using C, 3rd Ed, Oxford Press, 2012.
- 3) Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2 nd Ed, McGraw Hill, 2013.
- 4) A M Tenenbaum, Data Structures using C, PHI, 1989.
- 5) Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.

Course outcomes

CO #	Course Outcome
CO1	Implement dynamic memory operations and pointer-based structures in C programs.
CO2	Apply appropriate data structures and algorithms for organizing and processing data.
CO3	Use stacks, queues, linked lists, trees, and graphs for solving computational problems.
CO4	Implement and analyze data structures for problem-solving in a high-level programming environment.
CO5	Apply hashing techniques and collision-resolution strategies for efficient data retrieval.

OPERATING SYSTEM (Effective from the academic year 2021-2022) SEMESTER - III			
Course Code	21AM34	CIE Marks	50
Number of Lecture Hours/Week (L:T:P)	2:0:0	SEE Marks	50
Total Number of Lecture Hours	30	Exam Hours	3
CREDITS - 02			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> • Understand the core fundamental concepts of Operating Systems. • Differentiate between multiprocessing, multiprogramming, and multitasking • Apply various process CPU scheduling algorithms to manage processes. • Analyze deadlock identification and prevention mechanisms. • Apply segmentation and paging techniques 			
Module I			5 Hours
Introductions: Operating System Services, User- Operating System operations and interface, System Calls, Operating System design and implementation, Case Studies.			
			7 Hours
Processes & Thread Management: Process Overview, Process Scheduling algorithms, Inter-process communication. PCB, Multithreading models, Threading issues. Process Synchronization: The critical section problem, Peterson's solution, Mutex locks, Semaphores, and problems using synchronization.			
Module III			6 Hours
Deadlocks: System Model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock Avoidance, Deadlock detection, and recovery from deadlock.			
Module IV			6 Hours
Main Memory: Background, swapping, Contiguous memory allocation, Segmentation, Paging, Structure of page table.			
Virtual Memory: Background, Demand paging, Copy on write, Page replacement algorithms, Allocation of frames, Thrashing.			
Module V			6 Hours
Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Swap space management.			
Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix.			
Course Outcomes: After studying this course, students will be able to:			
<p>CO1: Explain the fundamental concepts and services of operating systems.</p> <p>CO2: Differentiate multiprocessing, multitasking, and multiprogramming environments.</p> <p>CO3: Apply CPU scheduling algorithms to evaluate process performance.</p> <p>CO4: Analyze deadlock conditions and propose avoidance or prevention mechanisms.</p> <p>CO5: Apply memory-management techniques such as segmentation and paging to solve allocation problems.</p>			
Question paper pattern:			
<ul style="list-style-type: none"> • The question paper will have ten questions. 			

- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1. Operating System Concepts, by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 9th Edition, Wiley India, 2012.
2. Operating systems, by H.M.Deitel, D.J.Deitel, D.R.Choffnes, 3rd edition, Pearson Education.

Reference Books:

1. Operating Systems, A Concept-Based Approach, by DM Dhamdhere, 3rd Edition, Tata Mcgraw-Hill, 2012.
2. Modern Operating Systems, by Andrew S. Tanenbaum and Herbert Bos, 4th Edition, Pearson, 2014.

Course outcomes

CO #	Course Outcome
CO1	Explain the fundamental concepts and services of operating systems.
CO2	Differentiate multiprocessing, multitasking, and multiprogramming environments.
CO3	Apply CPU scheduling algorithms to evaluate process performance.
CO4	Analyze deadlock conditions and propose avoidance or prevention mechanisms.
CO5	Apply memory-management techniques such as segmentation and paging to solve allocation problems.

PYTHON PROGRAMMING
(Effective from the academic year 2021-2022)

SEMESTER - III

Course Code	21AM35	CIE Marks	50
Number of Lecture Hours/Week(L:T:P)	3:0:0	SEE Marks	50
Total Number of Lecture Hours	40	Exam Hours	3

CREDITS - 03

Course Overview:

Python is a Programming Language that can be treated in a procedural way, an object-orientated way, or a functional way. It can be used on a server to create web applications, create workflows, connect to database systems, read and modify files, handle big data and perform complex mathematics. It can implement object-oriented features and exception handling; It can parse the strings using regular expressions. It can be used for implementing machine learning algorithms to develop solutions for interdisciplinary problems apart from any general problems leading to automation.

Course Learning objectives: This course will enable students to:

- Understand the syntax and semantics of the Python programming language.
- To illustrate the process of structuring the data using lists, and tuples.
- To inculcate knowledge of parsing of regular expressions and their usage in various application domains.
- Demonstrate the use of built-in functions to navigate the file system.
- Implement Object-Oriented Programming concepts in Python and discuss the files, Pandas, and Data Virtualization concepts.

Module I

8 Hours

Python Basics: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program, Flow control: Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit(), Functions: def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number

Module II

8 Hours

Lists: The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List, List-like Types: Strings and Tuples, References, Dictionaries, and Structuring Data: The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things

Module III

8 Hours

Manipulating Strings: Working with Strings, Useful String Methods, Project: Password Locker, Project: Adding Bullets to Wiki Markup Reading and Writing Files: Files and File Paths, The os.path Module, The File Reading/Writing Process, Saving Variables with the shelve Module, Saving Variables with the print.format() Function, Project: Generating Random Quiz Files, Project: Multiclipboard.

Module IV

8 Hours

Organizing Files: The shutil Module, Walking a Directory Tree, Compressing Files with the zip file Module, Project: Renaming Files with American-Style Dates to European-Style Dates, Project:

Backing Up a Folder into a ZIP File, Debugging: Raising Exceptions, Getting the Traceback as a String, Assertions, Logging, IDLE's Debugger.

Classes and objects: Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying, Classes, and functions: Time, Pure functions, Modifiers, Prototyping versus planning, Classes and methods: Object-oriented features, Printing objects, Another example, A more complicated example, The init method, The __str__ method, Operator overloading, Type-based dispatch, Polymorphism, Interface, and implementation.

Inheritance, Card objects, Class attributes, Comparing cards, Decks, Printing the deck, Add, remove, shuffle and sort, Inheritance, Class diagrams, Data encapsulation.

Module V

8 Hours

NumPy: Introduction to NumPy, Creating arrays, Indexing Arrays, Array Transposition, Universal Array Function, Array Processing, Array Input, and Output. Pandas and Data

Visualization: Introduction, Series and Data Frames in pandas and Data Visualization.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks: -

1. Learning to Program using Python by Cody Jackson, Second Edition, 2014.
2. Pandas for Everyone: Python Data Analysis by Daniel Y. Chen, First Edition, Pearson, 2018.
3. Python Data Science Handbook by Jake VanderPlas, O'Reilly, 2017,

Reference Books:

a.

Course outcomes

CO #	Course Outcome
CO1	Apply Python syntax, semantics, and control flow constructs for problem-solving.
CO2	Use lists, tuples, dictionaries, and other structures to organize and process data.
CO3	Apply regular expressions for string processing and text-based applications.
CO4	Use built-in modules to navigate directories, manage files, and perform system operations.
CO5	Implement object-oriented concepts, NumPy, Pandas, and visualization tools for analytical applications.

DATA STRUCTURES AND ITS APPLICATIONS USING C LAB

(Effective from the academic year 2021-2022)

SEMESTER - III

Course Code	21CSL36	CIE Marks	50
Number of Lecture Hours/Week	2	SEE Marks	50
Total Number of Lecture Hours		Exam Hours	3

CREDITS - 01

Course Learning objectives: This course will enable students to:

- To design, develop, test, and debug in C/C++ language considering appropriate data structure.
- Illustrate and implement data types such as stack, queue, and linked list and apply them to the given problem.
- Illustrate and implement the trees and other data structures.
- Implement all the programs in 'C / C++ Programming Language and Linux / Windows as OS.

Programs List:

1.	Design, Develop and Implement a menu-driven Program in C for the following array operations. <ol style="list-style-type: none"> a. Creating an array of N Integer Elements b. Display of array Elements with Suitable Headings c. Inserting an Element (ELEM) at a given valid Position (POS) d. Deleting an Element at a given valid Position (POS) e. Exit. Support the program with functions for each of the above operations.
2.	Design, Develop and Implement a Program in C for the following operations on Strings. <ol style="list-style-type: none"> a. Read the main String (STR), a. Pattern String (PAT), and a Replace String (REP) b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR Support the program with functions for each of the above operations. Don't use Built-in functions.
3.	Design, Develop and Implement a menu-driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) <ol style="list-style-type: none"> a. Push an Element on to Stack b. Pop an Element from Stack c. Demonstrate how Stack can be used to check Palindrome d. Demonstrate Overflow and Underflow situations on Stack e. Display the status of Stack f. Exit Support the program with appropriate functions for each of the above operations
4.	Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. The program should support both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power), and alphanumeric operands
5.	Design, Develop and Implement a Program in C for the following Stack Applications <ol style="list-style-type: none"> a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ b. Solving Tower of Hanoi problem with n disks
6.	Design, Develop and Implement a menu-driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) <ol style="list-style-type: none"> a. Insert an Element onto Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit Support the program with appropriate functions for each of the above operations

7.	Design, Develop and Implement a menu-driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Programme, Sem, Ph. No a. Create an SLL of N Students Data by using front insertion. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion / Deletion at End of SLL d. Perform Insertion / Deletion at Front of SLL(Demonstration of the stack) e. Exit
8.	Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion in Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit
9.	Design, Develop and Implement a Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes a. Represent and Evaluate a Polynomial $P(x,y,z) = 6x^2y^2z^4yz^5 + 3x^3yz + 2xy^5z - 2xyz^3$ b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) Support the program with appropriate functions for each of the above operations.
10.	Design, Develop and Implement a menu-driven Program in C for the following operations on the Binary Search Tree (BST) of Integers. a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder, and Post Order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
11.	Design, Develop and Implement a menu-driven Program in C for the following operations on the Binary Search Tree (BST) of Integers. a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder, and Post Order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
12.	Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L be Integers. Design and develop a Program in C that uses the Hash function $H: K \rightarrow L$ as $H(K) = K \text{ mod } m$ (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.
Course Outcomes: The student should be able to	
CO1: Analyze and compare linear and nonlinear data structures for problem-solving. CO2: Develop and debug programs using appropriate data structures. CO3: Implement, analyze, and evaluate searching and sorting algorithms. CO4: Select and apply suitable data structures for real-world applications.	
Conduct of Practical Examination:	
<ul style="list-style-type: none"> • Experiment distribution <ul style="list-style-type: none"> a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity. b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity. • Change of experiment is allowed only once and marks allotted for the procedure to be made zero of the changed part only. • Marks Distribution <ul style="list-style-type: none"> a) For laboratories having only one part - Procedure + Execution + Viva-Voce: 15+70+15 = 100 Marks b) For laboratories having PART A and PART B i. Part A - Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks ii. Part B - Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks 	

Course outcomes

CO #	Course Outcome
CO1	Analyze and compare linear and nonlinear data structures for problem-solving.
CO2	Develop and debug programs using appropriate data structures.
CO3	Implement, analyze, and evaluate searching and sorting algorithms.
CO4	Select and apply suitable data structures for real-world applications.

OPERATING SYSTEM LAB
(Effective from the academic year 2021-2022)

SEMESTER - III

Course Code	21AML37	CIE Marks	50
Number of Lecture Hours/Week	03	SEE Marks	50
Total Number of Lecture Hours		Exam Hours	3

CREDITS - 01

Course Learning objectives: This course will enable students to:

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management in contemporary OS
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
- To know the components and management aspects of concurrency management
- To learn programmatically to implement simple OS mechanisms

PART-A

1. Write a C program to Simulate the following CPU scheduling algorithms
a) FCFS b) SJF c) Round Robin d) Priority
2. Write a C program to Simulate IPC techniques
a) Pipes b) Message Queues c) Shared Memory
3. Write a C Program to Simulate Classical Problems of Synchronization
a) Readers-Writers b) Producers-Consumers c) Dining Philosophers
4. Write a C Program to simulate Bankers Algorithm for Dead Lock Avoidance
5. Write a C Programs to Simulate all page replacement algorithms
a) FIFO b) LRU c) Optimal Etc...
6. Write a C program to Simulate Disk Scheduling Algorithms
a) FCFS b) SSTF etc.

PART - B

1. Build a simple client-server program to transfer files from client to server.
2. Build a simple shell to transfer a file from the server and pipe it through a word count program.
3. Build a multithreaded server to transfer files from server to client. Each thread will read one file and send it back

Course Outcomes: After studying this course, students will be able to:

CO1: Simulate and evaluate different CPU scheduling algorithms.

CO2: Implement deadlock detection and avoidance algorithms.

CO3: Apply semaphore mechanisms to solve synchronization problems.

CO4: Create processes and implement inter-process communication (IPC) techniques.

CO5: Analyze the performance of various page-replacement algorithms.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Course outcomes

CO #	Course Outcome
CO1	Simulate and evaluate different CPU scheduling algorithms.
CO2	Implement deadlock detection and avoidance algorithms.
CO3	Apply semaphore mechanisms to solve synchronization problems.
CO4	Create processes and implement inter-process communication (IPC) techniques.
CO5	Analyze the performance of various page-replacement algorithms.

PYTHON PROGRAMMING LAB
(Effective from the academic year 2021-2022)

SEMESTER - III

Course Code	21AML38	CIE Marks	50
Number of Lecture Hours/Week	03	SEE Marks	50
Total Number of Lecture Hours		Exam Hours	03

CREDITS - 01

Course Learning objectives: This course will enable students to:

- Develop program to solve real world problems using python programming.
- Develop the programs using the concepts of control statements, data structures & files.
- Apply features of object-oriented and Numpy, pandas package to develop computationally intensive programming & interpret the data.

Part-A

No.	Title of the experiment	Expected Skill / Ability
1.	<p>a. Develop a program to read the students details like Name, USN, and Marks in three subjects. Display the student's details, total marks and percentage with suitable messages.</p> <p>b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.</p> <p>c. Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages.</p>	Basics of Python
2.	<p>a. Write a program to demonstrate different numbered data types in Python and perform different arithmetic operation on numbers in python.</p> <p>b. Write a program to create, concatenate and print a string and access a sub-string from a given string.</p> <p>c. Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017".</p> <p>d. Read a multi-digit number (as char) from the console. Develop a program to print the frequency of each digit with suitable messages.</p>	Basics of Python
3.	<p>a. Develop a program to find largest of three numbers</p> <p>b. Develop a program to generate Fibonacci sequence of length (N). Read N from the console.</p> <p>c. Write a program to calculate factorial of a number. Develop a program to compute binomial coefficient (Given N and R).</p>	Conditional Statements
4.	<p>a. Implementing programs using Functions (Largest number in a list, area of shape)</p> <p>b. implementing a real-time/technical application using Exception handling (Divide by Zero error, Voter's age validity, student mark range validation)</p>	Functions and Exception Handling

5.	<p>a. "LIST1" is a list that contain "N" different SRN of students read using a user-defined function with the help of input () function.it is required to add the SRN of "M" for more students that are to be appended or inserted into "LIST1" at the appropriate place. The program must return the index of the SRN entered by the user.</p> <p>b. TUPLE1" and "TUPLE2" are two tuples that contain "N" values of Different data types read using the user -defined function "READ" with the help of the input () function. Elements of "TUPLE1" and "TUPLE2" are to be read one at a time and the "larger" value among them should be placed into "TUPLE3". Display all tuples.</p>	<p>Create and Perform operations on the list</p> <p>Create and Perform operations on tuples</p>
6.	<p>a. SET1 and SET2 are two sets that contain unique integers. SET3 is to be created by taking the union or intersection of SET1 and SET2 using the user defined function Operation (). Perform either union or intersection by reading the choice from the user. Do not use built-in function union () and intersection () and also the operators " " and "&".</p> <p>b. The Dictionary "DICT1" contains N Elements and each element in the dictionary has the operator as the KEY and operands as VALUES. Perform the operations on operands using operators stored as keys. Display the results of all operations.</p>	<p>Create and Perform Union and intersection operations on sets</p> <p>Create Dictionary and operates using a user defined function</p>
7.	<p>a. Implementing programs using Strings. (Reverse, palindrome, character count, replacing characters)</p>	<p>String operations</p>
8.	<p>a. Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display dictionary slice of first 10 items]</p> <p>b. Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: String method strip(), list method sort(), append(), and file method open(), readlines(), and write()].</p> <p>c. Develop a program to backing up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.</p>	<p>File Handling</p>
9.	<p>a. Develop a program that uses class Students which prompts the User to enter marks in three subjects and calculate total marks, Percentage and display the score card details. [Hint: Use list to store the marks in three subjects and total marks. Use_init() method to initialize name, USN and the lists to store marks and total, Use getMarks() method to read marks into the list, and display() method to display the score card details.]</p>	<p>Classes and Objects usability</p>
10.	<p>a. Implementing program using modules and python Standard Libraries (pandas, Numpy, Matplotlib, Scipy)</p>	<p>pandas, Numpy, Matplotlib, Scipy Usability</p>

Course outcomes

CO #	Course Outcome
CO1	Develop Python programs for real-world computations and problem-solving.
CO2	Apply conditional statements, looping, functions, and exception handling effectively.
CO3	Use lists, tuples, sets, and dictionaries for structured data processing.
CO4	Apply file-handling techniques for data storage, retrieval, and manipulation.
CO5	Implement object-oriented concepts and use NumPy/Pandas for data analysis.

UNIX AND SHELL PROGRAMMING
(Effective from the academic year 2021-2022)
SEMESTER - III

Course Code	21AAM3111	CIE Marks	50
Number Lecture Hour/Week	01	SEE Marks	50
Number of Lecture Hours		Exam Hours	03

CREDITS-01

Introduction:

Unix Components/ Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Basic Unix commands such as echo, printf, ls, who, date, passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The root login. Becoming the super user: su command.

RBT: L1, L2

Unix files:

Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands.

Connecting commands:

Pipe. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions.

RBT: L1, L2

File attributes and permissions:

The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions.

RBT: L1, L2, L3

Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection.

Overview of IPC Methods,

Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores.

RBT: L1, L2, L3

Shared Memory,

Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions.

RBT: L1, L2, L3

Text Books:

1. Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill (Chapter 1,2 ,3,4,5,6,8,13,14)

2. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005 (Chapter 3,7,8,10,13,15)
3. Unix System Programming Using C++ - Terrence Chan, PHI, 1999. (Chapter 7,8,9,10)

Reference Books:

1. M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.
2. Richard Blum, Christine Bresnahan: Linux Command Line and Shell Scripting Bible, 2nd Edition, Wiley, 2014.

CO1	Utilize Basic UNIX commands effectively.
CO2	Write effective scripts for their day to jobs.
CO3	Debug and troubleshoot software issues effectively.
CO4	Analyze the data and interpret the results.
CO5	Prepare a well-organized laboratory report.

Scilab

(Effective from the academic year 2022-2023)

SEMESTER - III

Course Code	21AEC3112	CIE Marks	50
Teaching Hours/Week	02	SEE Marks	50
Total No .of Lecture hours	30	Exam Hours	03

CREDIT -01

Course Objectives:

- **Learn core skills of SCILAB**
- **Able to understand the basic concepts of programming**

1. Study of Basic Scilab Commands
2. Matrix Constructors and Operations
3. Matrix Bitwise, Relational & Logical Operations
4. Control Structures (If-Else, If-Elseif -Else, Select)
5. Control Structures (For, While, Break and Continue)
6. Graphics - 2D Plots
7. Scilab - Civil Application Program (1)
8. Scilab - Civil Application Program (2)
9. Scilab - Electronics Application Program (1)
10. Scilab - Electronics Application Program (2)

Conduct of Practical Examination:**• Experiment distribution**

- a) **For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.**
- b) **For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.**

• Change of experiment is allowed only once and marks allotted for procedure

• maximum marks. SEE for practical shall be evaluated for to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of 50 marks

CO#	COURSE OUTCOMES
CO1	Learn fundamental Scilab programming concepts, including matrix operations, control structures, and data visualization, to solve computational problems
CO2	Implement and demonstrate SCI lab basic programs
CO3	Debug and trouble shoot software issues effectively
CO4	Analyze the data and interpret the results
CO5	Prepare a well-organized laboratory report

ENGINEERING MATHEMATICS IV
(Effective from the academic year 2021-2022)
SEMESTER - IV

Course Code	21MAT41	CIE Marks	50
Number of Lecture Hours/Week (L: T:P)	3:0:0	SEE Marks	50
Total Number of Lecture Hours	40	Exam Hours	3
CREDITS - 03			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> • Learn Fourier series and Fourier transforms. • Conversant with numerical methods to solve ordinary differential equations. • Know then complex combers, Analytic function and associated results and problems. • Understand Joint probability distribution and stochastic processes arising in science and engineering. • Understand the definition of sequence, series and its importance. • Discuss the elementary concepts of graph theory. • Know the finite difference method and use in solving partial differential equation 			
Module I			8 hours
<p>Fourier Series: Periodic functions, Dirichlet's condition, Fourier Series of periodic functions with period 2π and with arbitrary period $2c$. Fourier series of even and odd functions Half range Fourier Series, practical harmonic analysis (5 Assignment Problem).</p> <p>Self-Study: Sequence and series of a function, convergent series. (RBT Levels: L1, L2 and L3)</p>			
Module II			8 hours
<p>Joint probability distribution: Joint Probability distribution for two discrete random variables, expectation, covariance, correlation coefficient.</p> <p>Stochastic process: Stochastic processes, probability vector, stochastic matrices, fixed points, regula r stochastic matrices, Markov chains, higher transition probability-simple problems. (5 Assignment Problem).</p> <p>Self Study : Joint probability distribution for continuous random variable (RBT Levels: L1, L2 and L3).</p>			
Module III			8 hours
<p>Numerical Methods: Numerical solution of ordinary differential equations of first order and first degree, Taylor's series method, modified Euler's-method Runge Kutta method of fourth order. Milne's and Adams- Bashforth predictor and corrector methods (No derivations of formulae). (5 Assignment Problem).</p> <p>Self Study : Picards method (RBT Levels: L1, L2 and L3).</p>			
Module IV			8 hours
<p>Numerical Methods: Numerical solution of second order ordinary differential equations, Runge- Kutta Method and Milne's Method, Numerical solution of P.D.E: Numerical solution of Heat equation, Wave equation, problems. (5 Assignment Problem).</p> <p>Self Study : Picard's method, Numerical solution of Laplace's equation (RBT Levels: L1, L2 and L3)</p>			
Module V			8 hours
<p>Graph Theory: introduction to graph theory, definitions of finite and null graphs, loops, multigraphs, pseudo graph, simple graph, degree of a vertex, isolated vertices, connectedness and complete graph, minimum and maximum degree, regular graphs, sub graphs, walk, trial, paths, Euler and Hamilton graphs.</p> <p>Self Study : line graphs , Bridge, Block and tree (RBT Levels: L1, L2 and L3)</p>			

Course Outcomes: After studying this course, students will be able to:

CO1: Apply Fourier series and transforms for analyzing periodic signals and engineering applications.

CO2: Analyze joint probability distributions and stochastic processes including Markov chains.

CO3: Apply single-step and multi-step numerical methods to solve first-order ODEs.

CO4: Use numerical techniques to solve second-order ODEs and PDEs arising in heat and wave equations.

CO5: Apply graph theory concepts to model and analyze real-world problems.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question -will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

3. B.S. Grewal : Higher Engineering Mathematics, Khanna Publishers, 43rd Ed., 2015.
2. E. Kreyszig : Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed.(Reprint), 2016.

Reference Books:

1. C.Ray Wylie, Louis C.Barrett : "Advanced Engineering Mathematics", 6th Edition, . McGraw-Hill Book Co., New York, 1995.
2. James Stewart : "Calculus -Early Transcendentals", Cengage Learning India Private Ltd., 2017.
3. B.V.Ramana : "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill,2010.
4. Srimanta Pal & Subobh C Bhunia:"Engineering Mathematics", Oxford University Press,3rd Reprint,2016.
5. Gupta C.B., Singh S.R. and Mukesh Kumar : "Engineering Mathematics for Semester I & II", Mc-Graw Hill Education (India) Pvt.Ltd., 2015

Course Outcome:

CO1	Apply Fourier series and transforms for analyzing periodic signals and engineering applications.
CO2	Analyze joint probability distributions and stochastic processes including Markov chains.
CO3	Apply single-step and multi-step numerical methods to solve first-order ODEs.
CO4	Use numerical techniques to solve second-order ODEs and PDEs arising in heat and wave equations.
CO5	Apply graph theory concepts to model and analyze real-world problems.

MICROCONTROLLER AND EMBEDDED SYSTEMS (Effective from the academic year 2021-2022) SEMESTER - IV			
Course Code	21AM42	CIE Marks	50
Number of Lecture Hours/Week (L:T:P)	3:0:0	SEE Marks	50
Total Number of Lecture Hours	40	Exam Hours	3
CREDITS - 03			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> • Understand the fundamentals of ARM-based systems, basic hardware components, selection methods, and attributes of an embedded system. • Program ARM controller using various instructions. • Identify the applicability of the embedded system. • Comprehend the real-time operating system used for the embedded system. 			
Module I			10 ours
<p>Preamble: This course is to make the students understand the architecture, programming, and interfacing of a system design of microprocessors and microcontrollers. Embedded systems are special-purpose computing systems embedded in application environments or in other computing systems and provide specialized support. The decreasing cost of processing power, combined with the decreasing cost of memory and the ability to design low-cost systems on a chip, has led to the development and deployment of embedded computing systems in a wide range of application environments.</p> <p>Microprocessors Versus Microcontrollers, ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software. ARM Processor Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table, Core Extensions.</p>			
Module II			10 Hours
<p>Introduction to the ARM Instruction Set: Data Processing Instructions, Program, Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, and Loading Constants. ARM programming using Assembly language: Writing Assembly code.</p>			
Module III			10Hours
<p>Embedded System Components: Embedded vs. General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, the purpose of embedded systems. Core of an Embedded System including all types of processors/controllers.</p>			
Module IV			10Hours
<p>Embedded System Design Concepts and its Components: Memory, Sensors, Actuators, LED, 7 segment LED display, stepper motor, Keyboard, Communication Interface (onboard and external types), Embedded firmware, Other system components. Characteristics and Quality Attributes of Embedded Systems, Operational quality attributes, non-operational quality attributes, Embedded Systems-Application and Domain specific.</p>			
Module V			10Hours

RTOS and IDE for Embedded System Design: Hardware Software Co-Design and Program. Modeling, embedded firmware design, and development, how to choose an RTOS, Embedded system Development Environment – Block diagram (excluding Keil), Disassembler / decompile.

Course Outcomes: After studying this course, students will be able to:

CO1: Explain the architecture, features, and instruction set of ARM microcontrollers.

CO2: Develop assembly and embedded C programs for ARM-based applications.

CO3: Identify and select hardware components based on embedded-system requirements.

CO4: Analyze real-time operating system concepts for embedded applications.

CO5: Apply hardware/software co-design principles to develop embedded firmware solutions.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question -will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

- 1) Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan Kaufman publishers, 2008.
- 2) Shibu KV, "Introduction to Embedded Systems", Tata Mc Graw Hill Education, Private Limited, 2nd Edition

Reference Books:

- 1) Raghunandan. G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication, 2019.
- 2) The Insider's Guide to the ARM7 Based Microcontrollers, Hitex Ltd., 1st edition, 2005.
- 3) Steve Furber, ARM System-on-Chip Architecture, Second Edition, Pearson, 2015.
- 4) Raj Kamal, Embedded System, Tata McGraw-Hill Publishers, 2nd Edition, 2008.

Course outcomes

CO #	Course Outcome
CO1	Explain the architecture, features, and instruction set of ARM microcontrollers.
CO2	Develop assembly and embedded C programs for ARM-based applications.
CO3	Identify and select hardware components based on embedded-system requirements.
CO4	Analyze real-time operating system concepts for embedded applications.
CO5	Apply hardware/software co-design principles to develop embedded firmware solutions.

DESIGN AND ANALYSIS OF ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2021-2022)

SEMESTER - IV

Course Code	21AM43	CIE Marks	50
Number Lecture Hour/Week	03	SEE Marks	50
Number of Lecture Hours	40	Exam Hours	03
CREDITS-03			
Course objectives: This course will enable students to			
<ul style="list-style-type: none"> • Explain various computational problem-solving techniques. • Apply appropriate method to solve a given problem. • Describe various methods of algorithm analysis. 			
Modules			Teaching Hours
Module -1			
Introduction: What is Algorithm? Notation of Algorithm, Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ) with examples, Mathematical analysis of non-Recursive and recursive Algorithms with Examples. Brute force design technique: Selection Sort, Bubble Sort, Sequential Search and Brute Force String Matching. Advantages and Disadvantages of Brute force design.			8
RBT: L1, L2, L3			
Module -2			
Divide and Conquer: General method, Recurrence equation for divide and conquer, solving it using Master's theorem, Merge sort, Quick sort, Binary search, Strassen's matrix multiplication, Advantages and Disadvantages of divide and conquer.			8
Decrease and Conquer Approach: Introduction, Insertion Sort, Depth-First Search and Breadth-First Search, Topological Sorting.			
RBT: L1, L2, L3.			
Module -3			
Space-Time Tradeoffs: Introduction, Sorting by Counting, Input Enhancement in String Matching Harspool's algorithm			8
Dynamic Programming: Transitive Closure: Warshall's Algorithm. All Pairs Shortest Paths: Floyd's Algorithm,			
Transform and Conquer Approach: Introduction, Heaps and Heap Sort.			
RBT: L1, L2, L3			
Module -4			
Greedy Method: Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm with performance analysis.			7
Single source shortest paths: Dijkstra's Algorithm.			
Optimal Tree problem: Huffman Trees and Codes.			
Backtracking: General method, N-Queens problem,			

RBT: L1, L2, L3.	
Module -5	
Backtracking: Sum of subsets problem, Hamiltonian cycles. Branch and Bound: Assignment Problem, Travelling Sales Person problem, 0/1 Knapsack problem NP-Complete and NP-Hard problems: P, NP, NP-Complete and NP-Hard classes RBT: L1, L2, L3.	7
Text Books: 1. Introduction to the design and analysis of algorithms, by Anany Levitin, 3rd Edition, Pearson Education, 2011. 2. Computer Algorithms, by Horowitz E., Sahani S., Rajasekharan S., 2nd Edition, Universities Press, 2008.	
Reference Books: 1. Introduction to Algorithms, Cormen T.H, Leiserson C. E, Rivest R.L, Stein C, 3rd Edition, PHI 2010. 2. Data Structures and Algorithm Analysis in C++, by Mark Allen Weiss, PHI, 2013.	

Course outcomes

CO #	Course Outcome
CO1	Analyze algorithms using asymptotic notations and recurrence relations.
CO2	Apply divide-and-conquer and decrease-and-conquer techniques for solving computational problems.
CO3	Use dynamic programming and transform-and-conquer methods to optimize problem-solving.
CO4	Apply greedy and backtracking strategies to design efficient algorithms.
CO5	Analyze NP-hard and NP-complete problems and apply branch-and-bound solutions where applicable.

INTRODUCTION TO ARTIFICIAL INTELLIGENCE			
(Effective from the academic year 2021-2022)			
SEMESTER - IV			
Course Code	21AM44	CIE Marks	50
Number of Lecture Hours/WeekWeek (L:T:P)	3:0:0	SEE Marks	50
Total Number of Lecture Hours	40	Exam Hours	3
CREDITS - 03			
Course Learning objectives: This course will enable students to:			
<ol style="list-style-type: none"> 1. Gain a historical perspective of AI and its foundations. 2. Become familiar with basic principles of AI toward problem solving 3. Get to know approaches of inference, perception, knowledge representation, and learning. 			
Module I			8 Hours
Introduction to AI: history, Intelligent systems, foundation and sub area of AI, applications, current trend and development of AI. Problem-solving: state space search and control strategies.			
Chapter 1 and 2			
RBT: L1, L2			
Module II			8 Hours
Problem reduction and Game playing: Problem reduction, game playing, Bounded look-ahead strategy, alpha-beta pruning, Two player perfect information games			
Chapter 3			
RBT: L1, L2			
Module III			8 Hours
Logic concepts and logic Programming: Propositional calculus, Propositional logic, natural deduction system, semantic tableau system, resolution refutation, predicate logic, Logic programming.			
Chapter 4			
RBT: L1, L2			
Module IV			8 Hours
Advanced problem solving paradigm: Planning: types of planning system, block world problem, logic based planning, Linear planning using a goal stack, Means-ends analysis, Nonlinear planning strategies, learning plans			
Chapter 6.			
RBT: L1, L2			
Module V			8 Hours
Knowledge Representation, Expert system: Approaches to knowledge representation, knowledge representation using semantic network, extended semantic networks for KR, Knowledge representation using Frames. Expert system: introduction phases, architecture ES verses Traditional system			
Chapter 7 and 8 (8.1 to 8.4)			
RBT: L1, L2			
Course Outcomes: After studying this course, students will be able to:			
<p>CO1: Describe fundamental concepts of AI and problem-solving approaches.</p> <p>CO2: Apply AI search strategies for solving constrained and unconstrained problems.</p> <p>CO3: Model intelligent agents and knowledge-representation structures.</p> <p>CO4: Apply reasoning techniques such as decision trees, logic, and uncertainty modeling.</p> <p>CO5: Evaluate AI applications across various domains.</p>			
Question paper pattern:			

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1. Saroj Kaushik, Artificial Intelligence, Cengage learning, 2014

Reference Books:

1. Elaine Rich, Kevin Knight, Artificial Intelligence, Tata McGraw Hill
2. Nils J. Nilsson, Principles of Artificial Intelligence, Elsevier, 1980
3. Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson Education, 3rd Edition, 2009
4. George F Luger, Artificial Intelligence Structure and strategies for complex, Pearson Education, 5th Edition, 2011.

COURSE OUTCOME:

CO#	Course Outcomes
CO1	Describe fundamental concepts of AI and problem-solving approaches.
CO2	Apply AI search strategies for solving constrained and unconstrained problems.
CO3	Model intelligent agents and knowledge-representation structures.
CO4	Apply reasoning techniques such as decision trees, logic, and uncertainty modeling.
CO5	Evaluate AI applications across various domains.

FOUNDATIONS OF DATA SCIENCE (Effective from the academic year 2021-2022)			
SEMESTER - IV			
Course Code	21AM45	CIE Marks	50
Number of Lecture Hours/Week (L:T:P)	2:0:0	SEE Marks	50
Total Number of Lecture Hours	40	Exam Hours	3
CREDITS - 03			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> To provide a comprehensive knowledge of data science using Python. To learn the essential concepts of data analytics and data visualization. 			
Module I			8 Hours
Data science: definition, Datafication, Exploratory Data Analysis, The Data science process, A data scientist role in this process. NumPy Basics: The NumPy ndarray: A Multidimensional Array Object, Creating ndarrays ,Data Types for ndarrays, Operations between Arrays and Scalars, Basic Indexing and Slicing, Boolean Indexing, Fancy Indexing, Data Processing Using Arrays, Expressing Conditional Logic as Array Operations, Methods for Boolean Arrays , Sorting , Unique			
Module II			8 Hours
Getting Started with pandas: Introduction to pandas, Library Architecture, Features, Applications, Data Structures, Series, DataFrame, Index Objects, Essential Functionality (Reindexing, Dropping entries from an axis, Indexing, selection, and filtering), Sorting and ranking, Summarizing and Computing Descriptive Statistics, Unique Values, Value Counts, Handling Missing Data, filtering out missing data.			
Module III			8 Hours
Data Loading, Storage, and File Formats : Reading and Writing Data in Text Format, Reading Text Files in Pieces, Writing Data Out to Text Format, Manually Working with Delimited Formats, JSON Data, XML and HTML: Web Scraping, Binary Data Formats, Using HDF5 Format, Reading Microsoft Excel Files, Interacting with Databases, Storing and Loading Data in MongoDB .			
Module IV			8 Hours
Data Wrangling: Combining and Merging Data Sets, Database style DataFrame Merges, Merging on Index, Concatenating Along an Axis, Combining Data with Overlap , Reshaping and Pivoting, Reshaping with Hierarchical Indexing, Data Transformation, Removing Duplicates, Replacing Values.			
Module V			8 Hours
Plotting and Visualization: A Brief matplotlib API Primer, Figures and Subplots, Colors, Markers, and Line Styles, Ticks, Labels, and Legends, Annotations and Drawing on a Subplot, Saving Plots to File, Plotting Functions in pandas, Line Plots, Bar Plots, Histograms and Density Plots, Scatter Plots.			
Course Outcomes: After studying this course, students will be able to:			
<p>CO1: Explain the role of data science and its lifecycle in real-world applications.</p> <p>CO2: Apply data preprocessing techniques for cleaning and preparing datasets.</p> <p>CO3: Use statistical methods to analyze and interpret data patterns.</p> <p>CO4: Apply data visualization tools for effective representation of insights.</p> <p>CO5: Use basic machine-learning methods for classification, clustering, and prediction.</p>			
Question paper pattern:			
<ul style="list-style-type: none"> The question paper will have ten questions. 			

- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1. Wes McKinney, "Python for Data Analysis", O'REILLY, ISBN:978-1-449-31979-3, 1st edition, October 2012.

2. Rachel Schutt & O'neil, "Doing Data Science", O'REILLY, ISBN:978-1-449-35865-5, 1st edition, October 2013

Reference Books:

1. Joel Grus, "Data Science from Scratch: First Principles with Python", O'Reilly Media, 2015

2. Matt Harrison, "Learning the Pandas Library: Python Tools for Data Munging, Analysis, and Visualization O'Reilly, 2016.

COURSE OUTCOME:

CO1	Explain the role of data science and its lifecycle in real-world applications.
CO2	Apply data preprocessing techniques for cleaning and preparing datasets.
CO3	Use statistical methods to analyze and interpret data patterns.
CO4	Apply data visualization tools for effective representation of insights.
CO5	Use basic machine-learning methods for classification, clustering, and prediction.

BLOOM'S Level of The Course Outcomes

CO#	Bloom's Level					
	Remember (L1)	Understand (L2)	Apply (L3)	Analyze (L4)	Evaluate (L5)	Create (L6)
CO1	√	√	√			
CO2	√	√	√	√		√
CO3	√	√	√	√		
CO4	√	√	√	√		
CO5	√	√	√	√		

Course Articulation Matrix

CO#/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO2
CO1	1				2						1	1	1		3
CO2	1		1				1				1		1	2	
CO3	2			1							1				
CO4	1	1	1		1					1			1	1	
CO5	3	1	1	2		1				1			2	1	

Note: 1-Low, 2-Medium, 3-High

DESIGN AND ANALYSIS OF ALGORITHMS LAB (Effective from the academic year 2021-2022) SEMESTER - IV			
Course Code	21AML46	CIE Marks	50
Number of Lecture Hours/Week	02	SEE Marks	50
Total Number of Lecture Hours	20	Exam Hours	3
CREDITS - 01			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> • Design and implement various algorithms in C • Employ various design strategies for problem solving. • Measure and compare the performance of different algorithms 			
PART-A			
1	Design a program to search a key element of n integers using binary search algorithm and compute time complexity		
2	Design a program to Sort a given set of n integer elements using Quick Sort method and compute its time complexity.		
3	Design a program to sort set of n integer elements using Merge Sort method and compute its time complexity.		
4	Implement the 0/1 Knapsack problem using (a) Dynamic Programming method. (b) Greedy method.		
5	Design a program to print all the node reachable from a given starting node in a given digraph using DFS method.		
PART-B			
1	Write a Program find shortest paths to other vertices using Dijkstra's algorithm.		
2	(a) Write a program to find a Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. (b) Write a program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.		
3	Write a program to (a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm. (b) Implement transitive closure using Warshall Algorithm..		
4	Design and implement to find a subset of a given set.		
5	Implement Travelling Salesman problem using Dynamic program		
Course Outcomes: After studying this course, students will be able to:			
CO1: Implement algorithms using various design techniques such as divide-and-conquer and dynamic programming.			
CO2: Analyze correctness and efficiency of algorithms using test cases.			
CO3: Apply algorithmic strategies to real-time computational problems.			
Conduct of Practical Examination:			
<ul style="list-style-type: none"> • Experiment distribution <ul style="list-style-type: none"> b) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity. c) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity. 			

- Change of experiment is allowed only once and marks allotted for the procedure to be made zero of the changed part only.
- Marks Distribution
 - a) For laboratories having only one part - Procedure + Execution + Viva-Voce: $8+35+7 = 50$ Marks
 - b) For laboratories having PART A and PART B
 - i. Part A - Procedure + Execution + Viva = $3 + 14+ 3 = 20$ Marks
 - ii. Part B - Procedure + Execution + Viva = $5 + 21 + 4 = 30$ Marks

COURSE OUTCOME:

CO1	Implement algorithms using various design techniques such as divide-and-conquer and dynamic programming.
CO2	Analyze correctness and efficiency of algorithms using test cases.
CO3	Apply algorithmic strategies to real-time computational problems.

ARTIFICIAL INTELLIGENCE LAB (Effective from the academic year 2021-2022) SEMESTER - IV			
Course Code	21AML47	CIE Marks	50
Number of Lecture Hours/Week	02	SEE Marks	50
Total Number of Lecture Hours		Exam Hours	3
CREDITS - 01			
Course Learning objectives: This course will enable students to:			
<ul style="list-style-type: none"> Implement and evaluate AI algorithms in Python programming language. 			
Programs List:			
Practicing Problems in Python(Students can be encouraged to practice good number of practice problems , some practice problems are listed here)			
PART - A			
1	(a) Write a python program to print the multiplication table for the given number (b) Write a python program to check whether the given number is prime or not? (c) Write a python program to find factorial of the given number?		
2	(a) Write a python program to implement List operations (Nested List, Length, Concatenation, Membership, Iteration, Indexing and Slicing) (b) Write a python program to implement List methods (Add, Append, and Extend & Delete).		
3	Write a python program to implement simple Chabot with minimum 10 conversations		
4	Write a python program to Illustrate Different Set Operations		
5	(a)Write a python program to implement a function that counts the number of times a string(s1) occurs in another string(s2) (b)Write a program to illustrate Dictionary operations([],in,traversal)and methods: keys(),values(),items()		
PART - B			
1	Implement and Demonstrate Depth First Search Algorithm on Water Jug Problem		
2	Implement and Demonstrate Best First Search Algorithm on any AI problem		
3	Implement AO* Search algorithm.		
4	Solve 8-Queens Problem with suitable assumptions		
5	Implementation of TSP using heuristic approach		
6	Implementation of the problem-solving strategies: either using Forward Chaining or Backward Chaining		
7	Implement resolution principle on FOPL related problems		
8	Implement any Game and demonstrate the Game playing strategies		
Course Outcomes: After studying this course, students will be able to:			
CO1: Implement AI search algorithms for problem-solving.			
CO2: Apply knowledge-representation techniques in AI applications.			
CO3: Develop programs demonstrating reasoning, inference, and decision-making.			
COURSE OUTCOME:			
CO#	Course Outcomes		
CO1	Implement AI search algorithms for problem-solving.		
CO2	Apply knowledge-representation techniques in AI applications.		
CO3	Develop programs demonstrating reasoning, inference, and decision-making.		

DATA SCIENCE LAB
(Effective from the academic year 2021-2022)

SEMESTER - IV

Course Code	21AML48	CIE Marks	50
Number of Lecture Hours/Week	03	SEE Marks	50
Total Number of Lecture Hours	48	Exam Hours	3

CREDITS - 01

Course Learning objectives: This course will enable students to:

- The main objective of the course is to inculcate the basic understanding of Data Science and its practical implementation using Python.

Programs List:

Practicing Problems in Python (Students can be encouraged to practice good number of practice problems , some practice problems are listed here)

1	Creating a NumPy Array of the following Basic ndarray, Array of zeros and ones, Random numbers in ndarray and Evenly spaced ndarray
2	Write a following programs for Shape and Reshaping of NumPy Array a. Dimensions of NumPy array b. Shape of NumPy array c. Size of NumPy array d. Reshaping a NumPy array e. Flattening a NumPy array f. Transpose of a NumPy array
3	Write a Python program for Indexing and Slicing of NumPy Array a. Slicing 1-D NumPy arrays b. Slicing 2-D NumPy arrays c. Slicing 3-D NumPy arrays d. Negative slicing of NumPy arrays
4	Expanding and Squeezing a NumPy Array a. Sorting in NumPy array b. Expanding of NumPy array c. Squeezing of NumPy Arrays
5	Write a python program for Concatenating and stacking of Numpy Arrays a. Stacking ndarrays b. Concatenating ndarrays c. Broadcasting in Numpy Arrays
6	Perform the following operations using pandas: Creating data frame, concat(), Setting conditions and Adding a new column
7	Perform following operations using pandas a. Filling NaN with string b. Sorting based on column values c. c. groupby()
8	Read the following file formats using pandas: Text files, CSV files, Excel files and JSON files.
9	Read the following file formats: Pickle files, Image files using PIL, Multiple files using Glob and Importing data from database
10	Demonstrate web scraping using python

11	Demonstration of Feature Scaling, Feature Standardization, Label and One Hot Encoding-processing technique on loan prediction dataset
12	Demonstration Pie Chart, Bar Graph, Line Chart, Histogram and Scaller Plot using matplotlib

Course Outcomes: After studying this course, students will be able to:

CO1: Apply data preprocessing operations on raw datasets.

CO2: Perform exploratory data analysis using statistical and visualization techniques.

CO3: Implement basic machine-learning algorithms using tools such as Python, Pandas, and NumPy.

COURSE OUTCOME:

CO1	Apply data preprocessing operations on raw datasets.
CO2	Perform exploratory data analysis using statistical and visualization techniques.
CO3	Implement basic machine-learning algorithms using tools such as Python, Pandas, and NumPy.

SCRIPTING LANGUAGES
(Effective from the academic year 2021-2022)
SEMESTER - IV

Course Code	21AAM4111	CIE Marks	50
Number of Lecture Hours/Week	01	SEE Marks	50
Total Number of Lecture Hours		Exam Hours	3

CREDITS - 01

Course objectives: This course will enable students to:

- Use JavaScript for dynamic effects
- To prepare PHP scripts
- Use JavaScript & PHP to validate form input entry

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.
2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
3. Write JavaScript to validate the following fields of the Registration page.
 - a. First Name (Name should contains alphabets and the length should not be less than 6 characters).
 - b. Password (Password should not be less than 6 characters length).
 - c. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
 - d. Mobile Number (Phone number should contain 10 digits only).
 - e. Last Name and Address (should not be Empty).
4. Write an HTML page including any required JavaScript that takes a number from text field in the range of 0 to 999 and shows it in words. It should not accept four and above digits, alphabets and special characters.
5. Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT-SHRINKING" in BLUE color. Then the font size decreases to 5pt.
- 6.a. Install and configure PHP, web server, MYSQL
 - b. Write a program to print "Welcome to PHP"
 - c. Write a simple PHP program using expressions and operators.
7. Develop and demonstrate PHP Script for the following problems:
 - a) Write a PHP Script to find out the Sum of the Individual Digits.
 - b) Write a PHP Script to check whether the given number is Palindrome or not
8. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
9. Write a PHP Program to display current Date, Time and Day.
10. Write the PHP programs to do the following:
 - a. Implement simple calculator operations.
 - b. Find the transpose of a matrix.
 - c. Multiplication of two matrices.
 - d. Addition of two matrices.
11. Write a PHP program named states.py that declares variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:
 - a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.

- b. Search for a word in states that begins with k and ends in s. Perform a case insensitive comparison. Store this word in element1 of statesList.
 - c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.
 - d. Search for a word in states that ends in a. Store this word in element 3 of the list.
12. Write a PHP program to sort the student records which are stored in the database using selection sort.
13. Write a PHP program for sending and receiving plain text message (email).

DATA ANALYTICS WITH EXCEL

Semester - IV

Course Code	21AAM4112	CIE Marks	50
Number of Lecture Hours/Week (L: T:P)	0:0:2	SEE Marks	50
Total Number of Lecture Hours	32	Exam Hours	03

Credit: 01

Course Learning objectives: This course will enable students to:

- Apply analysis techniques to datasets in Excel.
- Learn how to use Pivot Tables and Pivot Charts to streamline your workflow in Excel.
- Understand and Identify the principles of data analysis.
- Become adept at using Excel functions and techniques for analysis.
- Build presentation ready dashboards in Excel.

1. **Getting Started with Excel:** Creation of spread sheets, Insertion of rows and columns, Drag & Fill, use of Aggregate functions.
2. **Working with Data :** Importing data, Data Entry & Manipulation, Sorting & Filtering.
3. **Working with Data:** Data Validation, Pivot Tables & Pivot Charts.
4. **Data Analysis Process:** Conditional Formatting, What-If Analysis, Data Tables, Charts & Graphs.
5. **Cleaning Data with Text Functions:** use of UPPER and LOWER, TRIM function, Concatenate.
6. **Cleaning Data Containing Date and Time Values:** use of DATEVALUE function, DATEADD and DATEDIF, TIMEVALUE functions.
7. **Conditional Formatting:** formatting, parsing, and highlighting data in spreadsheets during data analysis.
8. **Working with Multiple Sheets:** work with multiple sheets within a workbook is crucial for organizing and managing data, perform complex calculations and create comprehensive reports.
9. **Create worksheet with following fields:** Empno, Ename, Basic Pay(BP), Travelling Allowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT), Provident Fund(PF), Net Pay(NP). Use appropriate formulas to calculate the above scenario. Analyse the data using appropriate chart and report the data.
10. Generation of report & presentation using Autofilter & Macro.

Course Outcomes: After studying this course, students will be able to:

- Use advanced functions and productivity tools to assist in developing worksheets.
- Manipulate data lists using Outline and PivotTables.
- Use Consolidation to summarise and report results from multiple worksheets.
- Apply Macros and Autofilter to solve the given real world scenario.

Suggested Learning Resources:

1. <https://www.udemy.com/course/data-analytics-in-excel/>
2. https://www.tutorialspoint.com/excel_data_analysis/excel_data_analysis_tutorial.pdf
3. <https://www.youtube.com/live/9AOHwmmq5ug?feature=shared>