Programme structure for B. Tech. Computer Science and Engineering

Second Year Computer Science and Engineering Programs Semester III: Teaching Scheme

Course code	Course name	Teaching scheme (Hrs/week)			Credits assigned			Total credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
U14BS301	Statistics and Numerical Techniques	3	-	-	3	-	-	3
U14PC302	Computer Organization & Architecture	3	-	-	3	-	-	3
U14PC303	Object Oriented Programming using C++	3	-	-	3	-	-	3
U14MM0X	Multidisciplinary Minor I	2	-	-	2	-	-	2
U14OE01X	Open Elective I	2	-	-	2	-	-	2
U01EM001	Engineering Economics	2	-	-	2	-	-	2
U14PC304	Computer Organization & Architecture Lab	-	2	-	-	1	-	1
U14PC305	Object Oriented Programming using C++ Lab	-	4	-	-	2	-	2
U01VE002	Environmental Sciences	2	-	-	2	-	-	2
U14FP001	Field Project I		4	-	-	2	-	2
Total		**17**	**10**	**-**	**17**	**5**	**-**	**22**

***Open Elective is selected from the bucket of Open Elective List:**

Second Year Computer Science and Engineering Programs Semester III: Evaluation Scheme

Course code	Course name	*Theory Marks*					*Practical Marks*		Total
		Internal Assessment				ESE	Term work	Practical Oral/POE	
		T1	T2	FET	Total				
U14BS301	Statistics and Numerical Techniques	10	10	5	25	50	-	-	75
U14PC302	Computer Organization & Architecture	10	10	5	25	50	-	-	75
U14PC303	Object Oriented Programming using C++	10	10	5	25	50	-	-	75
U14MM0X	Multidisciplinary Minor I	10	-	5	15	35	-	-	50
U14OE01X	Open Elective I	10	-	5	15	35	-	-	50
U01EM001	Engineering Economics	10	-	5	15	35	-	-	50
U14PC304	Computer Organization & Architecture Lab	-	-	-	-	-	25	-	25
U14PC305	Object Oriented Programming using C++ Lab	-	-	-	-	-	25	25	50
U01VE002	Environmental Sciences	10	-	5	15	35	-	-	50
U14FP001	Field Project I	-	-	-	-	-	25	25	50

* Minimum passing is 40% for all courses and evaluation head mentioned above. FET – Faculty evaluation for Theory , T1, T2, Continuous Assessment Test, Term Work , ESE - End Semester Examination, P/F – Pass/ Fail Course, AU – Audit Course

Index

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14BS301	Statistics and Numerical Techniques	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14BS301	Statistics and Numerical Techniques	10	10	5	25	40%	50	40%	75

Course Description: This course is at third semester of CSE. It is a foundation course in Mathematics and may be pre-requisites for other courses and next semester Mathematical subjects. It covers statistics and numerical techniques to solve nonlinear and simultaneous linear equations.

Pre-requisites: Applied Mathematics-I, Applied Mathematics-II

Course Objectives:

- To equip students with the analytical tools and methodologies necessary for effective data analysis.
- To equip students with the analytical tools and methodologies necessary for effective decision-making, and problem-solving.
- To equip students with to develop appropriate models for curve fitting to given datasets.
- To equip students with the theoretical understanding and practical skills necessary for effectively solving Algebraic, Transcendental Equations.
- To equip students with the theoretical understanding and practical skills necessary for effectively solving of linear and non linear equations using numerical methods.

Course Outcomes: after the end of this course students will be able to

CO1 **Apply**[3] measures of central tendency to interpret and summarize data accurately.

CO2 **Utilize**[3] probability distributions to solve problems in engineering and science.

CO3 **Estimating**[4] the appropriate curve to approximate the bivariate data.

CO4 **Solve**[3] non linear and system of linear equations numerically.

Course Contents

Module	Unit	Description	Hours
1.0		**Statistics**	
1	1.1	Introduction, Measures of central Tendency: Mean, Median, Mode. Dispersion.	9
	1.2	Measures of Dispersion: Range, Variance, And Standard Deviation, Data Visualization Techniques: Histograms, Box Plots and Scatter Plots.	
2.0		**Probability Distribution**	
2	2.1	Random variable, Discrete probability distribution, continuous probability distribution, Probability density function.	9
	2.2	Binomial distribution, Poisson distribution and Normal distribution.	
3.0		**Curve Fitting and lines of Regressions**	
3	3.1	Fitting of curves by method of least squares, Coefficient of correlation.	9
	3.2	Regression coefficients, relation between coefficient of regression and regression coefficient, Lines of regression of bivariate data.	
4.0		**Solution of Algebraic and Transcendental Equations**	
4	4.1	Introduction, Bracketing method, Roots of Equation by Bisection Method, False position method.	9
	4.2	Open method, Roots of Equation by Secant Method, Newton- Raphson method, multiple roots by Newton Raphson method.	
5.0		**Linear System Equations:**	
5	5.1	Direct Methods of Solution – Solution of Linear Systems by Matrix inversion method, Gaussian Elimination method, Gauss Jordan method.	9
	5.2	Iterative Methods of Solution-Gauss Seidal method, Jacobi iterative method.	

Text Books

1 Introduction to Probability and Statistics for Engineers andScientistsby Sheldon M. Ross

2 Computer Based Numerical and Statistical Techniques by Manish Goyal, Laxmi Publications (P) Ltd, Third edition.

References

1 Grewal. B. S "Higher Engineering Mathematics", 41st Edition, Khanna Publications, Delhi, (2011).

2 Dass, H.K., and Er. RajnishVerma, Higher Engineering Mathematics, S. Chand Private Ltd., (2011).

3 Glyn James, Advanced Modern Engineering Mathematics, 3rd Edition, Pearson Education, (2012).

4 Peter V and O'Neil, Advanced Engineering Mathematics, 7th Edition, Cengage learning, (2012).

Internal Assessment (T1, T2 and FET):

1. T1 (Test 1) should be based on first two modules and T2 (Test 2) should be based on next two modules, for 10 marks each.
2. Fifth module (MCQ quiz) will be assessed for 5 marks separately.

End Semester Examination:

1. Question paper will comprise of 5 questions, each carrying 10 marks.
2. The duration of end semester examination shall be 2 hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14PC302	Computer Organization & Architecture	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14PC302	Computer Organization & Architecture	10	10	5	25	40%	50	40%	75

Course Description

The course has been designed to introduce fundamental principles of Computer Organization and Microprocessor. The students completing this course will understand basic of Computer Organization and Microprocessor, including CPU, ALU, Control Unit, Memory hierarchy, Computer performance, pipelining, Instruction Set and addressing modes of 8086 and Assembly programming of 8086. Finally, students will gain experience in learning concepts of Computer Organization and Programing of 8086 microprocessor

Pre-requisites: U01ES004- Basic Electrical and Electronics Engineering and U01VS001- Workshop Practice-I: C Programming

Course Objectives

- To understand basic components of computers.
- To understand the architecture of 8086 processor
- To Understand the principles of hardwired and microprogrammed control, including microprogram sequencing in addressing it.
- To describe the basic concept and hierarchy of memory systems in digital computing
- To understand the parallelism both in terms of single and multiple processors.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Explain**[2] architectures of Microprocessors for demonstrating working of data, address and control bus by using its pin configuration.
- **CO2** **Describe**[2] an assembly language program for given problem statement using 8086 microprocessor instruction set.
- **CO3** **Explain**[2] the evolution of computers & computer organization basics for understanding of the components of the system.
- **CO4** **Illustrate**[3] Control design and memory organization for designing of the memory system by using independent memory chips.
- **CO5** **Explain**[4] concepts of parallel processing and vector processing architecture for designing of the parallel processors by using the pipeline architectures.

Course Contents

Module	Unit	Description	Hours
1.0		**8086 microprocessor**	**9**
1	1.1	Introduction to Microprocessors, Features, pin functions and internal architecture of 8086.Flag register, Memory segmentation, Segment Registers, Physical address - calculation with examples, Physical memory organization. Interfacing 8086 with memory and I/O devices under minimum mode (Block-diagram level), Comparison between Minimum mode and Maximum mode configuration	
	1.2	Addressing modes - with example, Role of index and pointer registers. 8086 instruction set- Data transfer, arithmetic, logical, shift and rotate, branching, loop control and string instructions, processor control instructions with simple examples.	
2.0		**8086 assembly language programming**	**9**
2	2.1	ALP program development cycle, development tools, TASM-Assembler directives, structure of assembly program, sample programs -with relevant comments- such as data transfer,	
	2.2	Code conversion, largest/smallest, sorting, and searching, string palindrome and other simple programs. Comparison of procedure and macro.	

3.0		**Basic Computer Organization**	**9**
3	3.1	Evolution of computers - Mechanical era, Electronic computers, CPU organization, Data representations, Instruction Sets, RISC & CISC, definition, comparison and examples.	
	3.2	Basic concepts, Hardwired control Unit, Micro-programmed control unit, Memory Technology, Memory Systems, Caches: Main features	
4.0		**Memory**	**9**
4	4.1	Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories.	
	4.2	Cache memories: concept and design issues & performance, address Mapping and replacement Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation.	
5.0		**Computer Arithmetic and Different parallel processing architectures**	**9**
5	5.1	Number representation : Signed Integers ,Fixed point numbers, Floating point numbers, Floating point arithmetic's: Floating point addition, other Floating point operations, Booth's Algorithm, IEEE Standards for Floating point representations (Single & Double Precision Format)	
	5.2	Different parallel processing architectures: Introduction to Associative memory processors, Principles of multithreading, Latency hiding techniques.	

Text Books

1. A. K.Ray , K M Bhurchandi, "Advanced Microprocessor & Peripherals", Tata McGraw Hill,3nd Edition,2013
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012
3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998

References

1. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006
2. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach",

Elsevier, a division of reed India Private Limited, Fifth edition, 2012

3. Douglas V Hall, "Microprocessor & Interfacing: Programming and Hardware", Tata McGraw Hill, 2nd Edition,2006

4. Computer System Architecture - M. Mano

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately..

End Semester Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14PC303	Object Oriented Programming using C++	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14PC303	Object Oriented Programming using C++	10	10	5	25	40%	50	40%	75

Course Description

This course Object-Oriented Programming using C++ is an essential course designed for second-year computer engineering students to master the principles and practices of object-oriented programming (OOP) using the C++ programming language. This course serves as a foundational building block for understanding modern software development methodologies and techniques.

Pre-requisites: U01VS001- Workshop Practice-I: C Programming

Course Objectives:

- To explore & understand the principles of Object Oriented Programming (OOP).
- To use the object-oriented paradigm in program design.
- To use Idea about interference, diffraction and polarisation etc.
- To provide object-oriented programming insight using C++
- To lay a foundation for advanced programming.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Apply**[3] the concept of class, object, array, pointers inheritance and polymorphism to solve mathematical problems using C++ programming language.
- **CO2** **Make**[3] use of the various library utilities and advanced features like Template, STL to execute and handle multiple programs using C++ programming language
- **CO3** **Develop**[3] application using Stream I/O and File I/O to perform read and write operations using C++ programming language.
- **CO4** **Develop**[3] application to solve real world problems by using C++ programming language.
- **CO5** **Analyze**[4] the strengths of object oriented programming.

.

Course Contents

Module	Unit	Description	Hours
1.0		**Fundamentals of Object Oriented Programming**	**9**
1	1.1	Introduction to procedural, modular, generic and object-oriented programming techniques, limitations of procedural programming, Need of object-oriented programming, OOP Paradigms, Fundamentals of object-oriented programming: Namespaces, objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism. Benefits of OOP, C++ as object oriented programming language	
	1.2	C++ Programming- C++ programming Basics, Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, Access specifiers, separating interface from implementation. Functions- Function, function prototype, accessing function and utility function, Constructors and destructor, Types of constructor, Objects and Memory requirements, Static members: variable and functions, inline function, friend function.	
2.0		**Inheritance and Pointers**	**9**
2	2.1	Inheritance- Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Public and Private	

		Inheritance, Types of Inheritance, Ambiguity in Multiple Inheritance, Virtual Base Class, Abstract class, Friend Class , Nested Class.	
	2.2	Pointers: declaring and initializing pointers, indirection Operators, Memory Management: new and delete, Pointers to Objects, this pointer, Pointers Vs Arrays, accessing Arrays using pointers, Arrays of Pointers, Function pointers, Pointers to Pointers, Pointers to Derived classes, Passing pointers to functions, Return pointers from functions, Null pointer, void pointer.	
3.0		**Polymorphism**	**9**
3	3.1	Polymorphism- Introduction to Polymorphism, Early and late binding, Types of Polymorphism: Operator Overloading- concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable. Function overloading	
	3.2	Run Time Polymorphism- Pointers to Base class, virtual function and its significance in C++, pure virtual function and virtual table, virtual destructor, abstract base class	
4.0		**Files and Streams and Exception Handling**	**9**
4	4.1	Data hierarchy, Stream and files, Stream Classes, Stream Errors, Disk File I/O with Streams, File Pointers, and Error Handling in File I/O, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Object, Command-Line Arguments, Printer output	
	4.2	Exception Handling- Fundamentals, other error handling techniques, simple exception handling- Divide by Zero, Multiple catching, re-throwing an exception, exception specifications, user defined exceptions, processing unexpected exceptions, constructor, destructor and exception handling, exception and inheritance.	
5.0		**Standard Template Library**	**9**
5	5.1	Templates- , The Power of Templates, Function template, overloading Function templates, and class template, class template and Nontype parameters, template and friends Generic Functions, The typename and export keywords.	

	5.2	Introduction to STL, STL Components, Containers- Sequence container and associative containers, container adapters, Application of Container classes: vector, list, Algorithms- basic searching and sorting algorithms, min-max algorithm, set operations, heap sort, Iterators- input, output, forward, bidirectional and random access. Object Oriented Programming – a road map to future	

Text Books

1. E Balagurusamy Object-Oriented Programming with C++.7th edition. McGraw-Hill Publication,
2. Robert Lafore, ― Object-Oriented Programming in C++‖, fourth edition, Sams Publishing.

References

1. Herbert Schildt, ―C++ The complete reference‖, Eighth Edition, McGraw Hill Professional.
2. Matt Weisfeld, ―The Object-Oriented Thought Process, Third Edition Pearson.
3. Cox Brad, Andrew J. Novobilski, ―Object –Oriented Programming: An EvolutionaryApproach‖, Second Edition, Addison–Wesley.
4. Deitel, "C++ How to Program", 4th Edition, Pearson Education.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM011	Data communication	02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM011	Data communication	10	-	5	15	40%	35	40%	50

Course Description

Data Communication is a foundational course in computer engineering that explores the principles, techniques, and technologies used to transmit, receive, and exchange data between devices and systems. This course covers various aspects of data communication, including transmission media, protocols, network architectures, and the OSI model. Students will learn about different networking technologies, such as Ethernet, TCP/IP, and wireless communication, and gain hands-on experience with networking hardware and software.

Pre-requisites: - Basic understanding of computer systems and networks.
Familiarity with programming concepts and languages.

Course Objectives

- Understand the fundamental principles of data communication and networking
- Identify different types of transmission media and their characteristics.
- Understand the network protocols and services, such as DHCP, DNS, and NAT.
- Explain the layers of the OSI model and their functions in data communication.
- Analyze and design network architectures based on specific requirements

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Identify**[2] different types of transmission media and their properties
- **CO2** **Describe**[2] the operation of various networking protocols, such as TCP/IP and Ethernet
- **CO3** **Implement**[3] network protocols and services, such as DHCP, DNS, and NAT.
- **CO4** **Implement**[3] a network routing Algorithms for a given scenario or requirement.
- **CO5** **Design**[4] and implement a network solution for a given scenario or requirement.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction to Data Communication**	**6**
1	1.1	Definition and Importance of Data Communication Components of Data Communication System Data Transmission Modes: Simplex, Half-Duplex, Full-Duplex.	
	1.2	Transmission Media: Guided (Twisted Pair, Coaxial Cable, Fiber Optic) and Wireless (Radio Waves, Microwaves, Infrared)	
2.0		**Networking Fundamentals**	**6**
2	2.1	Basics of Computer Networks, Network Topologies: Bus, Star, Ring, Mesh , Network Devices: Switches, Routers, Hubs, Modems	
	2.2	Network Protocols and Standards: TCP/IP, OSI Model, Local Area Networks (LANs), Wide Area Networks (WANs)	
3.0		**Data Link Layer**	**6**
3	3.1	Functions and Services of Data Link Layer, Error Detection and Correction Techniques and example solving.	
	3.2	Data Link Control Protocols: HDLC, PPP, Medium Access Control (MAC) Protocols: CSMA/CD, CSMA/CA, Ethernet Standards and Operation	
4.0		**Network Layer**	**6**
4	4.1	Role and Functions of Network Layer, Routing Algorithms: Distance Vector, Link State	
	4.2	Internet Protocol (IP) and IP Addressing, Subnetting and Supernetting,IPv4 vs. IPv6	
5.0		**Transport Layer and Application Layer**	**6**
5	5.1	Functions and Services of Transport Layer, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).	
	5.2	Domain Name System (DNS),Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP)	

Text Books

1. Data and Computer Communications" by William Stallings.
2. Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross

References

1. Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall
2. TCP/IP Illustrated, Volume 1: The Protocols" by Kevin R. Fall and W. Richard Stevens
3. Networking Essentials" by Microsoft Official Academic Course

Internal Assessment (T1 and FET)

1. T1 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM021	Introduction to Data Science and Analytics	02	00	-	02	--	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM021	Introduction to Data Science and Analytics	10	-	5	15	40%	35	40%	50

Course Description

This course introduces the scope of data science and analytics. Statistical fundamentals required for data science are introduced. Overview of tools for data science is given. Data science project life cycle is discussed. Exploratory Data Analysis and the Data Science Process are illustrated.

Pre-requisites: Basics of computer programming C/C++.

Course Objectives:

- Understand and describe the role of data science and its tools.
- Understand and describe the role of big data and cloud computing in data science.
- Provide an overview of machine learning techniques such as supervised learning, unsupervised learning, and semi-supervised learning
- Discuss ethical issues surrounding data science, including privacy, bias, and data misuse.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** Describe the role of data science and its tools**.**
- **CO2** Describe the role of big data and cloud computing in data science.
- **CO3** Apply mathematical and statistical principles to the analysis of data.
- **CO4** Apply the techniques of Exploratory Data Analysis.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction**	**6**
1	1	Introduction to Data Science – Evolution of Data Science – Data Science Roles – Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues.	
2.0		**Data Collection and Data Pre-Processing**	**6**
2	2	Data Collection Strategies – Data Pre-Processing Overview – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization.	
3.0		**Exploratory Data Analytics**	**6**
3	3	Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis – Box Plots – Pivot Table – Heat Map – Correlation Statistics – ANOVA.	
4.0		**Model Development**	**6**
4	4	Simple and Multiple Regression – Model Evaluation using Visualization – Residual Plot – Distribution Plot – Polynomial Regression and Pipelines – Measures for In-sample Evaluation – Prediction and Decision Making.	
5.0		**Model Evaluation**	**6**
5	5	Generalization Error – Out-of-Sample Evaluation Metrics – Cross Validation – Overfitting – Under Fitting and Model Selection – Prediction by using Ridge Regression – Testing Multiple Parameters by using Grid Search.	

Text Books

1.	Jojo Moolayil, “Smarter Decisions : The Intersection of IoT and Data Science”, PACKT, 2016.

References

1.	Cathy O’Neil and Rachel Schutt , “Doing Data Science”, O'Reilly, 2015.
2.	David Dietrich, Barry Heller, Beibei Yang, “Data Science and Big data Analytics”, EMC 2013
3.	Raj, Pethuru, “Handbook of Research on Cloud Infrastructures for Big Data Analytics”, IGI Global.

Internal Assessment (T1, T2 and FET)

1. T2 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM031	Privacy And Security in Online social media	02	00	-	02	--	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM031	Privacy And Security in Online social media	10	-	5	15	40%	35	40%	50

Course Description

This course provides an in-depth exploration of the privacy and security challenges inherent in online social media platforms. Students will examine the risks associated with sharing personal information online and learn strategies for protecting their privacy and security in the digital realm.

Pre-requisites: Basics of computer application.

Course Objectives:

- Identify common privacy risks associated with sharing personal information on social media.
- Investigate techniques for enhancing personal data protection and digital privacy awareness
- Understand the tactics used by malicious actors to exploit vulnerabilities in social media platforms.
- Advocate for privacy-conscious behaviors and responsible digital citizenship in online social networks.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** Define key concepts related to privacy and security in online social media.
- **CO2** Explore the historical development and evolution of privacy concerns in social media platforms.
- **CO3** Analyze the impact of privacy breaches and security incidents on individuals and society.
- **CO4** Explore best practices for detecting and responding to security threats in social media environments.

Course Contents

Module	Unit	Description	Hours
1.0		**Foundations of Privacy and Security in Social Media**	**6**
1	1.1	Definition of privacy and security in the context of online social media. Overview of the importance of privacy and security in digital communication platforms. Exploration of the evolution of privacy concerns and security threats in social media environments.	
	1.2	Examination of key historical events and milestones shaping privacy discourse in social media. Discussion of landmark cases and legislative developments related to online privacy rights. Analysis of societal attitudes towards privacy and the evolution of privacy norms in the digital age.	
2.0		**Identifying Risks and Threats in Social Media Platforms**	**6**
2	2.1	Exploration of common privacy risks associated with sharing personal information on social media platforms. Discussion of data collection practices, including tracking cookies, user profiling, and targeted advertising. Examination of privacy implications of user-generated content, such as photos, videos, and status updates.	
	2.2	Overview of security threats prevalent in social media environments, including identity theft, phishing attacks, and malware distribution. Analysis of tactics used by cybercriminals to exploit vulnerabilities in social media platforms. Case studies illustrating real-world examples of identity theft, phishing scams, and data breaches on popular social media sites.	
3.0		**Implementing Privacy Protection Strategies**	**6**
3	3.1	Examination of privacy settings and controls available on popular social media platforms, including Facebook, Twitter, and Instagram. Step-by-step guides on accessing and customizing privacy settings to manage visibility of personal information and control data sharing.	
	3.2	Introduction to proactive strategies for minimizing privacy risks in social media environments, such as limiting sharing of personal information and using pseudonyms. Discussion of the role of privacy-enhancing technologies, such as virtual private networks (VPNs) and ad blockers, in protecting online privacy.	
4.0		**Mitigating Security Threats in Social Media**	**8**
4	4.1	Exploration of common tactics used by malicious actors to exploit vulnerabilities in social media platforms, including account takeover, phishing, and social engineering. Case studies illustrating real-world examples of security threats in social media environments and their impact on users.	

	4.2	Overview of best practices for securing personal accounts on social media platforms, such as using strong and unique passwords, enabling two-factor authentication (2FA), and regularly updating privacy settings. Guidance on identifying and reporting suspicious activity, such as unauthorized access or suspicious messages.	
	4.3	Strategies for detecting signs of security threats in social media accounts, such as unusual login activity, unrecognized friend requests, or suspicious links. Step-by-step instructions on how to respond to security incidents, including reporting abusive behavior, blocking malicious users, and contacting platform support for assistance.	
5.0		**Promoting Privacy and Security Awareness**	**6**
5	5.1	Discussion on the importance of advocating for privacy-conscious behaviors and responsible digital citizenship in online social networks. Strategies for promoting privacy awareness among peers, family, and broader online communities through education and outreach initiatives.	
	5.2	Exploration of effective communication strategies for raising awareness of privacy and security issues in online communities, including social media groups, forums, and online events. Collaboration with peers to develop and disseminate educational materials, such as infographics, articles, and videos, on privacy and security best practices.	
	5.3	Examination of the role of education and policy initiatives in promoting privacy and security awareness at individual, organizational, and societal levels. Discussion on the importance of advocating for privacy-respecting policies and regulations to protect users' digital rights and mitigate privacy risks in online social media platforms.	

Text Books

1.	Privacy and Big Data" by Terence Craig and Mary E. Ludloff.
2.	The Privacy Engineer's Manifesto: Getting from Policy to Code to QA to Value by Michelle Dennedy, Jonathan Fox, and Thomas R. Finneran

References

1.	Privacy in Context: Technology, Policy, and the Integrity of Social Life by Helen Nissenbaum
2.	Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World" by Bruce Schneier
3.	Networks of Control: A Report on Corporate Surveillance, Digital Tracking, Big Data & Privacy" by Wolfie Christl and Sarah Spiekermann

Internal Assessment (T1, T2 and FET)

1. T2 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
6. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM061	Software Development	02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM061	Software Development	10	-	5	15	40%	35	40%	50

Course Description

Software Development is a foundational course in computer engineering that explores understanding of software engineering principles and practices, equipping them with the skills necessary to develop high-quality software solutions. The course covers a range of topics, from software design and development methodologies to testing and maintenance strategies. Through a combination of lectures, hands-on projects, and practical exercises, students will learn how to design, implement, and deploy software systems efficiently and effectively.

Pre-requisites: - Basic understanding of computer systems.
Familiarity with programming concepts and languages.

Course Objectives

- Understand the principles and concepts of software engineering.
- Apply software design principles and patterns to develop modular, maintainable, and scalable software systems.
- Understand the importance of code reusability, abstraction, and encapsulation.
- Learn effective software testing techniques, including unit testing, integration testing, and system testing.
- Understand the ethical and professional responsibilities of software developers, including issues related to privacy, security, and intellectual property rights.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Identify**[2] different types of Software Engineering Processes and their properties in software development practices
- **CO2** **Describe**[2] the Process Planning, Effort Estimation and Software Configuration Management in software development practices
- **CO3** **Implement**[3] programming Principles and Guidelines in software development practices.

- **CO4** **Implement**[3] testing Fundamentals for a given scenario or requirement**.**
- **CO5** **Analyze**[4] different Testing Techniques software development practices**.**

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction to Software Engineering**	**6**
1	1.1	Types of Software, Software Characteristics, Quality of a Good Software, Software Myths, Software Components, Software Crisis, Software Engineering: Definition, Challenges, Software Engineering Processes, Similarity and Differences from Convential Engineering Processes. s.	
	1.2	Software Development Life Cycle Models: Build and Fix Models, Waterfall Model, Prototyping Model, RAD Model Iterative Enhancement Model, Evolutionary Development Model and Spiral Model, WINWIN Spiral Model, Fourth Generation Technique	
2.0		**Planning a Software**	**6**
2	2.1	Process Planning, Effort Estimation: Uncertainties in Effort Estimation, Building Effort Estimation Models, A Bottom-Up Estimation Approach, COCOMO Model, Project Scheduling & Staffing: Overall Scheduling, Detailed Scheduling, Team Structure,	
	2.2	Software Configuration Management (SCM): - Baselines, Version Control, Change Control & Configuration Audit, Risk Management: Reactive and Proactive Risk Strategies, Software Risks, Risk Analysis, Identification, Projection, Assessment, Monitoring and Managing the Risk, RMMM Plan.	
3.0		**Software Requirements Analysis and Specification**	**6**
3	3.1	Software Requirements Analysis and Specification: Software Requirements: Need for SRS, Requirement Process, Problem Analysis: Informal & formal Approaches, Data Flow Modeling, Object Oriented Modeling, Prototyping, Requirements Specifications: Characteristics of an SRS, Components of SRS,	

		Specification Language, Structure of Requirement Document: IEEE Standards for SRS, Validation, Metrics. Designing and Coding: Designing: Function Oriented Design: Design Principles: Problem Partitioning and Hierarchy, Abstraction, Modularity, Top Down and Bottom-Up Strategies,	
	3.2	Module Level Concepts: Coupling, Cohesion; Structure Design Methodology, Verification, Introduction to Object Oriented Design & User Interface Design, Software Measurement Metrics: Various Size Oriented Measures- Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.	
4.0		**Coding**	**6**
4	4.1	Coding: Programming Principles and Guidelines: Common Coding Errors, Structured Programming, Information Hiding, Programming Practices, Coding Standards, Coding Process,	
	4.2	Refactoring, Verification: Code Inspection, Static Analysis, Proving Correctness, Combining Different Techniques, Metrics.	
5.0		**Testing**	**6**
5	5.1	Testing Fundamentals: Error Fault and Failure, Test Oracles, Test Cases and Test Criteria, Test Case Execution and Analysis, Unit Testing, Integration Testing: : Top Down and Bottom up, Acceptance Testing: Alpha and Beta Testing., Regression Testing, functional and non-functional testing.	
	5.2	Testing Techniques: White Box: Logic Coverage, Path Coverage, Loop Coverage, Data Flow Testing. Black Box Testing: Boundary Value Analysis, Equivalence Class Testing, state Table Based Testing, Decision Table Based Testing.	

Text Books

1. Software Engineering: A Practitioner's Approach by Roger S. Pressman, McGraw-Hill International edition.
2. An Integrated Approach to Software Engineering, by Pankaj Jalote, Narosa Publishing House.

References

1. Software Engineering by K.K. Agarwal.
2. Software Engineering by Ian Sommerville, Addison-Wesley.
3. Fundamentals of Software Engineering by Rajib Mall, PHI

Internal Assessment (T1 and FET)

1. T1 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

***

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14OE011	Introduction to Latex	01	02	-	01	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
U14OE011	Introduction to Latex	50	40%	-	-	50

Course Description

This course is at even semester of first year Technology. The main motive is to impart knowledge and understanding about the LaTeX system, explain the procedure of LaTeX typesetting, and familiarize the participants with various document formats of LaTeX, enabling them to prepare articles, thesis, books, and presentations confidently.

Pre-requisites: Microsoft (MS) office.

Course Objectives

- Typesetting of complex mathematical formulae using LaTeX.
- To include figures and tables in a Latex document.
- Use the preamble of LaTeX file to define document class and layout options.
- Gaining hands-on experience in becoming a user of LaTeX

Course Outcomes: After the end of this course students will be able to

- **CO1** **Make use of** [3] Latex programming and commands, sample packages**.**
- **CO2** **Create**[6] well-written short documents and typeset it in LaTex.
- **CO3** **Create**[6] or import graphics into a LaTeX document.
- **CO4** **Create**[6] a simple report for a given technical topic, and typeset it in LaTex

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction**	**7**
1	1.1	Introduction to LaTeX, its installation, different commands and IDEs.	
	1.2	Creating the first document using LaTeX, organizing content into sections using article and book class of LaTeX.	
2.0		**Styling Pages**	**8**
2	2.1	Reviewing different paper sizes, examining packages, formatting the page by setting margins.	
	2.2	Customizing header and footer, changing the page orientation, dividing the document into multiple columns. Reading different types of error messages	
3.0		**Formatting Content**	**7**
3	3.1	Formatting text (styles, size, alignment), adding colours to text and entire page, and	
	3.2	Adding bullets and numbered items, writing complex mathematics. Inserting hyperlink, Inserting References, Inserting the Bibliography Styles	
4.0		**Tables and Images:**	**8**
4	4.1	Creating basic tables, adding simple and dashed borders, merging rows and columns.	
	4.2	Handling situations where a table exceeds the size of a page, adding an image, exploring different properties like rotate, scale.	

Suggested List of Experiments

1. Introduction to LaTeX, Creating and typesetting a simple LaTeX document.
2. Adding basic information to documents, Environments, Footnotes, Sectioning, Displayed material.
3. Accents and symbols; Mathematical typesetting (elementary and advanced): Subscript/ Superscript, Fractions, Roots, Ellipsis.
4. Mathematical symbols, Arrays, Delimiters, Multiline formulas
5. Putting one thing above another, Spacing and changing style in math mode.
6. Create project certificate. Features to be covered:-Formatting Fonts in word, Drop Cap in word, Applying

Text effects.

7. Creating a Newsletter: Features to be covered: - Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs.
8. Draft a CV using LaTeX templates.

Text Books

1. Guide to LATEX - UC Davis Mathematics, by H Kopka · 2004
2. The Not So Short Introduction to LaTeX 2E by Tobias Oetiker, Hubert Partl, https://cslab.pepperdine.edu/warford/cosc320/lshort.pdf

References

1. The LaTeX Companion Second Edition by Frank Mittelbach and Michel Goossens
2. Guide to LATEX, fourth edition, Helmut Kopka, Patrick W.Daly

Evaluation Scheme

1. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.
2. The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U01EM001	Engineering Economics	02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U01EM001	Engineering Economics	-	10	5	15	40%	35	40%	50

Course Description

The course focuses on economic of engineering projects, giving insights on different techniques and methods used on economic feasibility studies relating to design and implementation of engineering projects. The basic purpose of this course is to provide a sound understanding of concepts and principles of engineering economy and to develop proficiency with methods for making rational decisions regarding problems likely to be encountered in professional practice.

Pre-requisites: - mathematics

Course Objectives

- Understand the importance of economy in engineering field.
- Apply knowledge of economic comparisons for selecting best alternative
- Understand the concept of depreciation
- Explain importance of working capital management
- **Course Outcomes:** After the successful completion of the course students will able to:

- **CO1** **Explain**[2] importance of economy in engineering field.
- **CO2** **Correlate**[3] knowledge of economy in decision making
- **CO3** **Correlate**[3] knowledge of deprecation for engineering field
- **CO4** **Explain**[2] importance of working capital management in business.

Course Contents

Module	Unit	Description	Hours
1.0		**Fundamentals Economy**	**06**
1	1.1	Introduction to Engineering Economy, Time value of money, Cash flow, cash flow diagrams, simple Interest and compound Interest, inflation, economic factors.	
2.0		**Comparisons of Alternatives**	**06**
2	2.1	Present worth comparisons, Comparisons of assets with equal, unequal lives, comparison of deferred investments , Equivalent uniform annual cost (EUAC) method , Rate of return method , Future worth method, NPV method	
3.0		**Comparisons of Alternatives**	**06**
3	3.1	Payback period method- conventional and discounted payback period method, Benefit cost ration methods , break even analysis method	
4.0		**Depreciation and Inflation**	**06**
4	4.1	Definition method for calculating depreciation straight line method., constant percentage method, sinking fund method, Switching between different depreciation methods, inflation effect of inflation	
5.0		**working capital management**	**06**
5	5.1	Introduction, need of working capital management, financial ratio, Income statement, Financial statements	

References	
1	Leland Blnak, Anthony Tarquin, Engineering Economy , Tata McGraw Hill Publishing Company, New Delhi,
2	Jha, Kumar Neeraj., Construction Project management, Theory & Practice, Pearson Education India, 2015.
3	Panneer Selvam, R, "Engineering Economics", Prentice Hall of India Ltd, New Delhi, 2001
4	Degarmo, E.P., Sullivan, W.G and Canada, J.R, "Engineering Economy", Macmillan, New York, 2011

Internal Assessment (T1, T2 and FET)

3. T2 should be based on First to Fourth modules, for 10 marks.
4. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 07 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14PC304	Computer Organization & Architecture Lab	-	02	-	-	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term work	Min pass	Marks	Min pass	Total (Marks)
U14PC304	Computer Organization & Architecture Lab	25	40%	-	-	25

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Demonstrate**[3] use of assembly language programming for 8085 microprocessor by using 8086 TASM/MASM compiler.
- **CO2** **Design** [4]Arithmetic logic units and different types of memory blocks and organization of memory system for extending memory capacity using independent memory chips.

Suggested List of Experiments

All the experiments will be based on the course content of **Computer Organization & Architecture U14PC302**

1. Introduction to 8086 16-bit Microprocessor (Study Experiment)
2. Programs based on Arithmetic Operations of two 32 bit Numbers of 8086 Microprocessor.
3. Programs based on Logical Operations of 8086 Microprocessor
4. Program based on Branching Operations of 8086 Microprocessor
5. Program based on Conditional CALL and RET of 8086 Microprocessor using Simulator.
6. Program on data transfer from one Block to another block of Memory
7. Program based on interfacing between 8086 Microprocessor and I/O devices for designing interface structure.
8. Program based on Stack and subroutine of 8086 Microprocessor
9. Case study on Designing of a Memory system using Multiple Memory Independent Chips
10. Case study on Demonstration of Parallel Processors using Pipeline architectures

Text Books

1. A. K.Ray , K M Bhurchandi, "Advanced Microprocessor & Peripherals", Tata McGraw Hill,3nd Edition,2013
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012
3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998

References

1. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006
2. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012
3. Douglas V Hall, "Microprocessor & Interfacing: Programming and Hardware", Tata McGraw Hill, 2nd Edition,2006
4. Computer System Architecture - M. Mano

Evaluation Scheme

1. TERM WORK assessment shall be based on the overall performance of the student with every assignment graded from time to time.
2. The grades will be converted to marks as per 'credit and grading system' manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
U14PC305	Object Oriented Programming using C++ Lab	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
U14PC305	Object Oriented Programming using C++ Lab	Term Work	Min pass	Marks	Min pass	Total (Marks)
		25	40%	25	40%	50

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Write**[2] compile and debug programs in C++ language.
- **CO2** **Design**[4] programs involving object-oriented concepts for real word problem solving.

List of Experiments

All the experiments will be based on the course content of **Object Oriented Programming using C++**

U14PC303

EXERCISE-1(BASICS)

A. Write a CPP Program to demonstrate the structure of a C++ program.

B. Write a CPP Program to display the names of header files, definitions and list of functions supported.

C. Write a program to show the base of a numeric value of a variable using Hex, Oct and Dec manipulator functions.

D. Write a CPP Program to use of the standard manipulators normally used in the stream classes.

E. Write a CPP Program to demonstrate the usage of bit fields.

F. Write a CPP Program to define constant pointer and pointer to constant and perform possible operations.

G. Write a CPP Program access a variable in different scopes by using scope resolution operator and the use of comma operator.

EXERCISE-2(CLASSES & OBJECTS)

A. Write a CPP Program to swap two numbers using call by value, call by address, call by reference and return by reference.

B. Write a CPP Program to calculate square and cube of a number using inline functions and macros. (Demonstrate the use of inline functions compared to macros).

C. Write a CPP Program to find the area of a rectangle, a triangle and surface area of a sphere using function overloading.

D. Write a CPP Program to declare all members of a class as public, Access the members using objects. (Use public, protected, private).

E. Write a CPP Program to access the member functions inside and outside a class.

F. Write a CPP Program to access private data using non-member functions. (Use friend function).

G. Write a CPP Program to pass objects to functions by pass by value method.

H. Write a CPP Program to declare main () function as member function and overload it.

EXERCISE-3(CONSTRUCTORS AND OPERATOR OVER LOADING)

A. Write a CPP Program to show that "for each object constructors is called separately" and read the values through keyboard (Use Constructor).

B. Write a CPP Program to create constructor with arguments and pass the arguments to constructor.

C. Write a CPP Program to create object and release them using destructor.

D. Write a CPP Program to perform addition, subtraction, multiplication of two objects using operator keyword.

E. Write a CPP Program to overload unary and binary operator overloading with friend function.

EXERCISE-4(INHERITANCE AND POLYMORPHISM)

A. Write a CPP Program to derive a class publicly from base class. Declare base class members under public, private and protected.

B. Write a CPP Program to derive single and multiple inheritances.

C. Write a CPP Program to declare virtual base class. Derive a class using two virtual classes.

D. Write a CPP Program to implementation of Virtual Function.

E. Write a CPP Program to Implementation of Pure Virtual Function.

EXERCISE- 5(FILES, TEMPLATES AND EXCEPTION HANDLING)

A. Write a CPP Program to write and read text in a file. Use ofstream and ifstream classes.

B. Write a CPP Program to open a file for writing and reading purpose. Use open () function.

C. Write a CPP Program write text in a file. Read the text from the file from EOF. Display the contents in reverse order.

D. Write a CPP Program to demonstrate that the data is read from file using ASCII format.

E. Write a CPP Program to find the factorial of a number. Throw multiple exceptions and define multiple catch statements to handle exceptions.

F. Write a C++ Program to illustrate template class.

Text Books

1. E Balagurusamy Object-Oriented Programming with C++.7th edition. McGraw-Hill Publication
2. Robert Lafore, ― Object-Oriented Programming in C++‖, fourth edition, Sams Publishing.

References

1. Herbert Schildt, ―C++ The complete reference‖, Eighth Edition, McGraw Hill Professional, 2011.
2. Matt Weisfeld, ―The Object-Oriented Thought Process, Third Edition Pearson.
3. Cox Brad, Andrew J. Novobilski, ―Object –Oriented Programming: An EvolutionaryApproach‖, Second Edition, Addison–Wesley.
4. Deitel, "C++ How to Program", 4th Edition, Pearson Education.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
U01VE002	Environmental Sciences	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		2	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
U01VE002	Environmental Sciences	T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
		10	-	5	15	40%	35	40%	50

Course Description

This course will focus on basic concepts of Environmental components and basic concepts in the field. There will be a discussion about all the environmental conditions, problems, their Solutions, and policies regarding public awareness.

Pre-requisites: -

Course Objectives

- To Study of factors affecting the availability of natural resources, their conservation and management.
- Develop a critical understanding of the environmental issues of concern
- Study of the sectoral effects on the local, regional, and global environmental issues
- Study of the concepts of ecosystems, biodiversity and conservation.
- Study of factors impacting biodiversity loss and ecosystem degradation in India and the world.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Classify**[3] natural resources and their conditions in local area.
- **CO2** **Interpret2** local environmental issues
- **CO3** **Describe**[2] and modify solution on local environmental issues
- **CO4** **Develop**[4]a critical understanding of the complexity of environmental management.
- **CO5** **Explaining**[2] local common biodiversity

Course Contents

Module	Unit	Description	Hours
1.0		**Humans and the Environment**	**06**
1	1.1	Multidisciplinary nature of environmental studies; Scope and importance UN Conference on Human Environment 1972; World Commission on Environment and Development, the concept of sustainable development	
	1.2	**Natural Resources and Sustainable Development** Land resources and land use change; Land degradation, soil erosion and desertification. Deforestation: Causes and impacts due to mining, dam building on environment and forests. Water: Use and over-exploitation of surface and ground water, floods, droughts. Energy resources: Renewable and non-renewable energy sources, use of alternate energy sources, rowing energy needs, case studies	
2.0		**Environmental Issues: Local, Regional and Global**	**06**
2	2.1	Global issues: Climate change, global warming, ozone layer depletion, acid rain and impacts on human communities and agriculture.	
	2.2	**Conservation of Biodiversity and Ecosystems:** • Levels of biological diversity: genetic, species and ecosystem diversity; • Global biodiversity hot spots. India as a mega-biodiversity nation; Endangered and endemic species of India • Threats to biodiversity: Habitat loss, poaching of wildlife, man wildlife conflicts, biological invasions • Conservation of biodiversity In-situ and Ex-situ conservation of biodiversity. • Ecosystem and biodiversity services: Ecological, economic, social, ethical, Aesthetic, and Informational value	
3.0		**Climate Change: Impacts, Adaptation and Mitigation**	**06**

3	3.1	Environmental pollution: types, causes, effects and controls; Air, water, Noise pollution • Nuclear hazards and human health risks. Solid waste management: Control measures of urban and industrial waste	
	3.2	Structure of atmosphere; Anthropogenic climate change from greenhouse gas emissions– past, present and future; Mitigation of climate change	
4.0		**Environmental Management**	**06**
4	4.1	**Environment Laws**: Environment Protection Act; Air (Prevention & Control of Pollution) Act; Water (Prevention and control of Pollution) Act; Wildlife Protection Act Environmental management system: ISO 14001, Environmental audit and impact assessment; Ecolabeling /Ecomark scheme.	
	4.2	**Environmental Treaties and Legislation:** Major International organizations and initiatives: United Nations Environment Programme (UNEP), International Union for Conservation of Nature (IUCN), Intergovernmental Union panel on climate change (IPCC)	
5.0		**working capital management**	**06**
5	5.1	Environmental movements: Chipko, Silent valley, Bishnois of Rajasthan, Bhopal Disaster	
	5.2	Field visits to identify local/regional environmental issues, make observations including data collection and prepare a brief report.	
	5.3	Documentation of campus biodiversity.	

	5.4	Campus environmental management activities such as solid waste disposal, water management, and sewage treatment.	

Text Books

1. Gadgil, M., & Guha, R. 1993. This Fissured Land: An Ecological History of India. Univ. of California Press.
2. Gleeson, B. and Low, N. (eds.) 1999. Global Ethics and Environment, London, Rout- ledge.
3. Gleick, P. H. 1993. Water in Crisis. Pacific Institute for Studies in Dev.,Environment & Security. Stockholm Env. Institute, Oxford Univ. Press.
4. Jadhav, H & Bhosale, V.M. 1995. Environmental Protection and Laws. Himalaya Pub. House, Delhi

References

1. Harper, Charles L. (2017) Environment and Society, Human Perspectives on environmental Issues 6th Edition. Routledge
2. Jackson, A. R., & Jackson, J. M. (2000). Environmental Science: The Natural Environment and Human Impact. Pearson Education.
3. Pittock, Barrie (2009) Climate Change: The Science, Impacts and Solutions. 2nd Edition. Routledge.
4. Sharma B.K., 2001. Environmental Chemistry. Geol Publ. House, Meerut
5. Chiras, D. D and Reganold, J. P. (2010). Natural Resource Conservation: Management for a Sustainable Future.10th edition, Upper Saddle River, N. J.Benjamin/Cummins/Pearson.

Internal Assessment (T1, T2 and FET)

1. T2 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 07 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14FP001	Field Project I	-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
U14FP001	Field Project I	25	40%	25	40%	50

Course Description

The objective of this course is to provide second-year engineering students with a solid foundation in statistical analysis and numerical techniques, enabling them to apply these methods effectively in engineering problem-solving and decision-making contexts.

Pre-requisites: Basics Engineering.

Course Objectives

- To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problem.
- To emphasizes learning activities that are long-term, inter-disciplinary and student-centric.
- To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism
- To develop an ecosystem this may promote entrepreneurship and research culture among the students.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Solve**[3] real life problems by applying knowledge.
- **CO2** **Analyze**[4] alternative approaches, apply and use most appropriate one for feasible Solution.
- **CO3** **Classify**[4] software applications and identify unique features of various domains.
- **CO4** **Implement**[3] computer engineering skills to meet challenges in the real world.

Course Contents

Preamble:

Field Project I is an instructional approach designed to give students the opportunity to develop knowledge and skills through engaging projects set around challenges and problems they may face in the real world. FIELD PROJECT I, is more than just projects. With FIELD PROJECT I students "investigate and respond to an authentic, engaging, and complex problem, or challenge" with deep and sustained attention. FIELD PROJECT I is "learning by doing." The truth is, many in education are recognizing we live in a modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. (Reference: Wikipedia). Project based learning will also redefine the role of teacher as mentor in learning process. Along with communicating knowledge to students, often in a lecture setting, the teacher will also to act as an initiator and facilitator in the collaborative process of knowledge transfer and development. The FIELD PROJECT I model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It Brings what students should academically know, understand, and be able to do and requires students to present their problems, research process, methods, and results.

FIELD PROJECT I requires regular mentoring by faculty throughout the semester for successful completion of the idea/project tasks selected by the students per batch. For the faculty involved in FIELD PROJECT I , teaching workload of 4 Hrs/week/batch needs to be considered. The Batch should be divided into sub-groups of 4 to 5 students. Idea implementation /Real life problem/Complex assignments / activities / projects. Under project based learning is to be carried throughout semester and Credit for FIELD PROJECT I has to be awarded on the basis of internal continuous assessment and evaluation at the end of semester

Group Structure:

Working in supervisor/mentor monitored groups; the students plan, manage, and complete a task/project/activity which addresses the stated problem.

1. There should be team/group of 4-5 students

2. A supervisor/mentor teacher assigned to individual groups

Selection of Project/Problem:

The problem-based project oriented model for learning is recommended. The model begins with the identifying of a problem, often growing out of a question or “wondering”. This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated interdisciplinary or subject frame. A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students’ wondering within different disciplines and professional environments.

A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases. By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

There are no commonly shared criteria for what constitutes an acceptable project.

Projects vary greatly in the depth of the questions explored, the clarity of the learning goals, the content, and structure of the activity.

• A few hands-on activities that may or may not be multidisciplinary.

• Use of technology in meaningful ways to help them investigate, collaborate, analyse, synthesize, and present their learning.

• Activities may include- Solving real life problem, investigation, /study and Writing reports of in depth study, field work.

Assessment:

- The institution/head/mentor is committed to assessing and evaluating both student performance and program effectiveness.
- Progress of FIELD PROJECT I is monitored regularly on weekly basis. Weekly review of the work is necessary. During process of monitoring and continuous assessment and evaluation of the individual and the team performance is to be measured. FIELD PROJECT I is monitored and continuous assessment is done by supervisor /mentor and authorities.
- Students must maintain an institutional culture of authentic collaboration, self-motivation, peer-learning and personal responsibility. The institution/department should support students in this regard through guidance/orientation programs and the provision of appropriate resources and services. Supervisor/mentor and Students must actively participate in assessment and evaluation processes.
- Group may demonstrate their knowledge and skills by developing a public product and/or report and/or presentation.

1. Individual assessment for each student (Understanding individual capacity, role and involvement in the project)

2. Group assessment (roles defined, distribution of work, intra-team communication and togetherness)

3. Documentation and presentation

Evaluation and Continuous Assessment:

It is recommended that all activities should to be recorded regularly, regular assessment of work need to be done and proper documents need to be maintained at college end by both students as well as mentor (FIELD PROJECT I work book).Continuous Assessment Sheet (CAS) is to be maintained by all mentors/department and institutes.

Recommended parameters for assessment/evaluation and weightage:

1. Idea Inception and Awareness /Consideration of -Environment/ Social /Ethics/ Safety measures/Legal aspects (10%)

2. Outcomes of FIELD PROJECT I/ Problem Solving Skills/ Solution provided/ Final product (Individual assessment and team assessment) (40%)

3. Documentation (Gathering requirements, design &modelling, implementation/execution, use of technology and final report, other documents) (15%)

Text Books:

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). 2017

2. Problem Based Learning. byMahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary MargaretCapraro

Reference Books:

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core text book, 2 Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden

Evaluation Scheme

1. TERM WORK assessment must be based on the overall performance of the student with every assignment done on computer graded from time to time.
2. The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Programme structure for B. Tech. Computer Science and Engineering
Second Year Computer Science and Engineering Programs Semester IV: Teaching Scheme

Course code	Course Name	Teaching scheme (Hrs/week)			Credits assigned			Total credits
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	
U14PC401	Data Structure	3	-	-	3	-	-	3
U14PC402	Operating System	3	-	-	3	-	-	3
U14PC403	Database Management System	2	-	-	2	-	-	2
U14MM0XX	Multidisciplinary Minor II	2	-	-	2	-	-	2
U14OE02XX	Open Elective II	2	-	-	2	-	-	2
U01EM002	Engineering Management	2	-	-	2	-	-	2
U14VS401	Software Proficiency I (Java Programming)	-	4	-	-	2	-	2
U14PC404	Data Structure Lab	-	2	-	-	1	-	1
U14PC405	Database Management System Laboratory	-	2	-	-	1	-	1
U14AE403 U14AE404	Modern Indian Language – Hindi OR Modern Indian Language – Sanskrit	2	-	-	2	-	-	2
U03VE404	Universal Human Values II	2	-	-	2	-	-	2
Total		18	8		18	4		22

Second Year Computer Science and Engineering Programs Semester IV: Evaluation Scheme

Course code	Course Name	*Theory Marks*					*Practical Marks*		Total
		Internal Assessment				ESE	Term work	Practical Oral /POE	
		T1	T2	FET	Total				
U14PC401	Data Structure	10	10	5	25	50	-	-	75
U14PC402	Operating System	10	10	5	25	50	-	-	75
U14PC403	Database Management System	10	10	5	25	50	-	-	75
U14MM0XX	Multidisciplinary Minor II	10	-	5	15	35	-	-	50
U14OE02XX	Open Elective II	10	-	5	15	35	-	-	50
U01EM002	Engineering Management	10	-	5	15	35	-	-	50
U14VS401	Software Proficiency I (Java Programming)	-	-	-	-	-	25	25	50
U14PC404	Data Structure Lab	-	-	-	-	-	25	-	25
U14PC405	Database Management System Laboratory	-	-	-	-	-	25	-	25
U14AE403 U14AE404	Modern Indian Language – Hindi OR Modern Indian Language – Sanskrit	10	-	5	15	35	-	-	50
U03VE404	Universal Human Values II	10	-	5	15	35	-	-	50

* Minimum passing is 40% for all courses and evaluation head mentioned above. FET – Faculty evaluation for Theory, T1, T2, Continuous Assessment Test, Term Work, ESE - End Semester Examination, P/F – Pass/ Fail Course, AU – Audit Course

Exit Option:

- Students will have the flexibility to enter a programme in odd semesters and exit a programme after the successful completion of even semesters as per their future career needs.
- Students exiting the Second Year Programme after securing minimum 80credits will be awarded UG Diploma in the relevant Discipline /Subject provided they secure additional 8 credits in skill-based vocational courses (skill-based courses, internship, mini projects etc) offered during summer vacation after the second year.

Index

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
U14PC401	Data Structure	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
U14PC401	Data Structure	T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
		10	10	5	25	40%	50	40%	75

Course Description

The Data Structures course provides students with a foundational understanding of organizing and managing data efficiently in computer systems. Students will explore various data structures and algorithms used to store, retrieve, and manipulate data, gaining insight into their strengths, weaknesses, and applications. Through a combination of theoretical concepts and practical implementations, students will develop the skills necessary to analyze problems and design efficient data structures and algorithms for solving them.

Pre-requisites: U14PC303- Object Oriented Programming using C++.

Course Objectives

- To learn the principles behind the implementation, operation, and manipulation of each data structure.
- To develop the ability to analyze the time and space complexity of algorithms.
- To solve a variety of problems using data structure concepts, including sorting, searching, traversal, and manipulation.
- To develop skills in designing efficient algorithms and data structures that can be scaled and optimized for large-scale software projects.

Course Outcomes: After the end of this course students will be able to

- **CO1** **Describe**[2] fundamentals in data structures for solving problems using a programming language.
- **CO2** **Explain**[2] the fundamental concepts of structuring, managing and organizing the data for solving problems using linear data structures with ADTs.
- **CO3** **Explain**[2] the fundamental concepts of structuring, managing and organizing the data for solving problems using non-linear data structures with ADTs.

- **CO4** **Apply**[3] appropriate non-linear data structure to solve the problem using a programming language.
- **CO5** **Compare**[4] different data structure algorithms and searching, sorting methods for solving problems using complexity methods.

Course Contents

Module	Unit	Description	Hours
1.0		**Basics of Data Structures**	**9**
1	1.1	Introduction: Data and Information, Data Structure, Classification of Data Structures, Primitive Data Types, Abstract Data Types, Data structure vs. File Organization, Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm, Asymptotic Analysis and Notations, Big O Notation, Big Omega Notation, Big Theta Notation, Rate of Growth and Big O Notation.	
	1.2	Array: Introduction, One Dimensional Array, Memory Representation of One Dimensional Array, Traversing, Insertion, Deletion, Searching, Sorting, Merging of Arrays, Multidimensional Arrays, Memory Representation of Two Dimensional Arrays, General Multi-Dimensional Arrays, Sparse Arrays, Sparse Matrix, Memory Representation of Special kind of Matrices, Advantages and Limitations of Arrays.	
2.0		**Linked List**	**9**
2	2.1	Linked List, One-way Linked List, Traversal of Linked List, Searching, Memory Allocation and De-allocation, Insertion in Linked List, Deletion from Linked List, Copying a List into Other List, Merging Two Linked Lists, Splitting a List into Two Lists, Reversing One way linked List, Circular Linked List, Applications of Circular Linked List	
	2.2	Two way Linked List, Traversing a Two way Linked List, Searching in a Two way linked List, Insertion of an element in Two way Linked List, Deleting a node from Two way Linked List, Header Linked List	
	2.3	Applications of the Linked list, Representation of Polynomials, Storage of Sparse Arrays, Implementing other Data Structures.	

3.0		**Stack and Queue**	**9**
3	3.1	Stack: Introduction, Operations on the Stack Memory Representation of Stack, Array Representation of Stack, Applications of Stack, Evaluation of Arithmetic Expression, Matching Parenthesis, infix and postfix operations, Recursion.	
	3.2	Queue: Introduction, Queue, Operations on the Queue, Memory Representation of Queue, Array representation of queue, Linked List Representation of Queue, Circular Queue, Some special kinds of queues, Deque, Priority Queue, Application of Priority Queue, Applications of Queues.	
4.0		**Searching and Hashing**	**9**
4	4.1	Bubble, Selection, Insertion, Merge Sort. Heap, Memory Representation of Heap, Operation on Heap, Heap Sort. Searching: Sequential, Binary, Indexed Sequential Searches, Binary Search.	
	4.2	Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and rehashing.	
5.0		**Nonlinear Data Structure**	**9**
5	5.1	Tree: Tree, Binary Tree, Properties of Binary Tree, Memory Representation of Binary Tree, Operations Performed on Binary Tree, Reconstruction of Binary Tree from its Traversals, Huffman Algorithm, Binary Search Tree, Operations on Binary Search Tree, Advanced Tree Structures: Red Black Tree, Operations Performed on Red Black Tree, AVL Tree, Operations performed on AVL Tree, 2- 3 Tree, B-Tree.	
	5.2	Graph: Introduction, Graph, Graph Terminology, Memory Representation of Graph, Adjacency Matrix Representation of Graph, Adjacency List or Linked Representation of Graph, Operations Performed on Graph, Graph Traversal, Applications of the Graph, Reachability, Shortest Path Problems, Spanning Trees.	

Text Books

1. Data Structures- A Pseudocode Approach with C by Richard F. Gilberg and Behrouz A.Forouzon , Cengage Learning ,2 edition ,2004.
2. Data Structures with C Schaum's Outlines Series by S. Lipschutz Tata McGraw-Hil, 2017.
3. Data Structure using C by Reema Thareia, Oxford, 2 edition ,2014.

References

1. Data Structure using C A. M. Tanenbaum,Y. Langsam, M.J. Augenstein Prentice- Hall Of India Pvt. Limited 2003
2. Understanding Pointers in C Yashavant Kanetkar BPB Publication I 2009

Internal Assessment (T1, T2 and FET)

1. T1 (Test 1) should be based on first two modules and T2 (Test 2) should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14PC402	Operating System	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14PC402	Operating System	10	10	5	25	40%	50	40%	75

Course Description

The Operating Systems course introduces second-year computer engineering students to the fundamental concepts, principles, and functionalities of operating systems. Operating systems serve as the bridge between computer hardware and software applications, managing system resources and providing a user-friendly interface for efficient interaction with the computer. This course covers topics such as process management, memory management, file systems, device management, and security.

Pre-requisites: U14PC302- Computer Organization & Architecture.

Course Objectives

- To Gain a solid understanding of the fundamental principles and components of operating systems.
- To apply theoretical knowledge to practical scenarios through hands-on exercises and projects, developing proficiency in implementing basic operating system functionalities.
- To Develop problem-solving skills by analyzing and resolving issues related to process scheduling, memory allocation, file management, and device I/O operations.
- To understand the challenges of concurrent execution and synchronization in operating systems.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Explain**[2] basic concepts of operating system and their structures to compare various operating systems using various OS parameters.
- **CO2** **Analyze**[4] issues related to process scheduling and resource management with the help of different scheduling algorithm.

- **CO3** **Develop[4]** appropriate solution to solve critical section problem by using accurate operating system algorithm.
- **CO4** **Use[3]** deadlock handling and Memory management techniques with suitable method to handle a deadlock and memory management.
- **CO5** **Explain[4]** concepts of file handling and I/O subsystem for accessing a file as well as I/O devices using file and I/O controller.

Course Contents

Module	Unit	Description	Hours
1.0		**Overview**	**9**
1	1.1	Introduction to Operating Systems, Operating System structure, Types of Operating Systems, Operating System Services, Views of Operating System	
	1.2	System calls, Types of system Calls, System programs, Kernel-Types of kernel, Overview of Linux and Android OS	
2.0		**Process Management**	**9**
2	2.1	Process concept: Basic concepts, Process States, Process Control Block, Context switch, Operations on processes, Inter-process communication, Threads	
	2.2	Process Scheduling: Scheduling criteria, Types of Scheduler, Scheduling algorithms, Multiple-Processor scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling	
3.0		**Process Synchronization**	**9**
3	3.1	Background, the critical section problem, Peterson's solution, synchronization hardware,	

	3.2	Semaphores, classic problems of Synchronization, Monitor	
4.0		**Deadlock**	**9**
4	4.1	System model, deadlock characterization, methods for handling deadlocks,	
	4.2	Deadlock preventions, deadlock avoidance, deadlock detection, deadlock recovery.	
5.0		**Memory Management and Storage Management& I/O Subsystem**	**9**
5	5.1	Memory Management Strategies: Background, swapping, contiguous memory allocation, paging, structure of the page table, Segmentation. Virtual Memory Management: Background, demand paging, copy-on-write, page replacement, Thrashing	
	5.2	File System: File concept, access methods, Directory and disk structure, Disk Scheduling, file-system mounting, file sharing, protection, Overview of I/O system, I/O hardware, Application I/O interface, Kernel I/O subsystem.	

Text Books

1. Operating System Concepts Gagne Silberschatz, Galvin,John Wiley 9 edition 2009
2. Operating Systems - A Concept Based approach Dhananjay M Dhamdhere Tata McGraw Hill 4 edition 2007.
3. Operating Systems: Principles and Practice Thomas Anderson and Michael Dahlin Recursive Books 2 edition.

References

1. Operating System A Design Oriented Approach Charles Crowley Tata McGraw Hill 3 edition .
2. Operating System with Case Studies in Unix, Netware and Windows NT Achyut S. Godbole Tata McGraw Hill 5 edition 2007.
3. Operating Systems: Internals and Design Principles William Stallings Pearson Education International 8 edition 2014
. Ubuntu: The Complete Reference Richard Petersen McGraw-Hill Education 1 edition 2008.

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately..

End Semester Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
U14PC403	Database Management System	Theory	Practical	Tutorial	Theory	Practical	Tutorial
		02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
U14PC403	Database Management System	T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
		10	10	5	25	40%	50	40%	75

Course Description

This course provides second-year computer engineering students with a comprehensive understanding of Database Management Systems (DBMS). Through theoretical lectures and practical exercises, students will learn the fundamental concepts, principles, and techniques used in the design, implementation, and management of databases.

Pre-requisites: Basics of programming C and C++.

Course Objectives:

- To Understand the fundamental concepts of database management systems (DBMS) .
- To Gain proficiency in relational data modelling techniques, including entity-relationship (ER) modelling.
- To apply normalization principles to design efficient and scalable
- To Develop practical skills in using Structured Query Language (SQL) for database querying
- To Explore advanced database concepts such as transaction management, concurrency control, and data integrity mechanisms to ensure reliable and secure data management

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Design[4]** an ER diagram and relational schema to solve given problem using integrity constraints and normalization techniques.
- **CO2** **Apply[3]** the concepts of database system, conceptual database design, relational algebra, SQL, normalization to solve the given problems through designing the database.
- **CO3** **Apply[3]** concepts transaction processing and concurrency control to improve the security and system performance using transaction management, concurrency control and recovery techniques

- **CO4** **Demonstrate**[3] concepts of indexing, concurrency protocols and recovery algorithms to solve real world problems using DBMS concepts.
- **CO5** **Formulate**[6] the queries to perform the create, delete, extract and update operations on the database using structured query language.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction to databases and ER Model**	**6**
1	1.1	Introduction: General introduction to database systems, its advantages and applications, View of Database – Levels of data abstraction, Data models, Database languages, Database System Architecture, Database users and Administrator	
	1.2	ER Model: Entity set, Entity types, attributes, Notations, Relationship sets, Relationship types, Keys- super key, candidate key, primary key, Extended Features of ER Model-Generalization, Specialization and aggregation	
2.0		**Relational Model and SQL Relational Model**	**9**
2	2.1	Structure of Relational Database, Reduction of ER model into Relational schemas, Schema-instance distinction, Referential integrity and foreign keys, Pure languages, Relational algebra, Example queries	
	2.2	SQL: Introduction to SQL, Data definition statements with constraints, Insert, Update and Delete, Set Operations, Aggregate functions group by and having clauses, Nested Queries, Views, Complex Queries, Joins.	
3.0		**Functional Dependency and Normalization**	**9**
3	3.1	Importance of a good schema design, Motivation for normal forms, Atomic domains and1NF, Dependency theory - functional dependencies, Closure of a set of FD's,	
	3.2	Definitions of 2NF, 3NF and BCNF, Decomposition algorithms and desirable properties of them, Multivalued dependencies and 4NF, Join dependencies and Definition of 5NF.	

4.0		**Data Storage & Indexing**	**9**
4	4.1	File organization, Organization of records in files, Data Dictionary, Database Buffer	
	4.2	Indexing: Concept, Ordered Indices-Primary, Secondary, Multilevel, B+ Tree Index, Hashing, Hash Indices, Dynamic hashing, Multiple key access, Bitmap Indices	
5.0		**Transaction Management & Concurrency Control**	**9**
5	5.1	Transaction Processing: Concept, ACID properties, Transaction states, Storage Structure, Implementation of atomicity, isolation and durability, Serializability, Testing of Serializability.	
	5.2	Concurrency Control: Lock-based protocols, Timestamp - based Protocols, Validation -based Protocols, Multiple Granularities, Deadlock handling.	

Text Books

1. A Database system concepts A. Silberschatz, H.F. Korth, S.Sudarsha McGraw Hill Education 6 edition 2011.
2. Database Systems- A practical approach to Design, Implementation Thomos Connolly, Carolyn Begg Pearson Education. 4 edition 2009
3. Database Systems – Design, Implementation and Management Rob & Coronel Thomson Course Technology 5 edition 2008
4. Database Management Systems Raghu Ram Krishnan McGraw Hill 3edition 2002.

References

1. Fundamentals of Database Systems Ramez Elmasri and Shamkant Navathe Pearson Education 4 edition 2007.
2. Systems: Design, Implementation and management Peter Rof, Carlos Coronel Cengage Learning 7 edition 2014.
3. Principles of Database Systems J. D. Ullman Galgotia publications 1 edition 2011
4. SQL: A Complete Reference Alexis Leon, Mathews Leon McGraw Hill Education 1 edition 2002

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.

2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks.
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM012	Linux and Shell Programming	02	--	-	02	--	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM012	Linux and Shell Programming	10	-	5	15	40%	35	40%	50

Course Description

This course offers second-year computer engineering students a comprehensive understanding of the Linux operating system and shell programming. It equips students with the necessary skills to navigate the Linux environment efficiently, automate tasks using shell scripts, and perform system administration duties.

Pre-requisites: Basics of computer system.

Course Objectives:

- To familiarize students with the Linux operating system environment, including its architecture, file system structure, and command-line interface.
- To learn how to write, execute, and debug shell scripts for automating repetitive tasks and system administration.
- To understand file permissions and ownerships, and how to manage them effectively for security purposes.
- To Gain proficiency in text processing utilities and regular expressions for data manipulation and analysis.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Describe**[2] the architecture and features of UNIX Operating System and distinguish it from other Operating System
- **CO2** **Demonstrate**[3] UNIX commands for file handling and process control
- **CO3** **Write**[2] Regular expressions for pattern matching and apply them to various filters for a specific task
- **CO4** **Analyze**[4] a given problem and apply requisite facets of SHELL programming in order to devise a SHELL script to solve the problem

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction**	**6**
1	1.1	Introduction to Unix-Brief History-What is Unix-Unix Components-Using Unix-Commands in Unix-Some	
	1.2	Basic Commands-Command Substitution-Giving Multiple Commands.	
2.0		**File system**	**6**
2	2.1	The File system –The Basics of Files-What's in a File-Directories and File Names-Permissions- INodes-The Directory Hierarchy,	
	2.2	File Attributes and Permissions-The File Command knowing the File Type-The Chmod Command Changing File Permissions-The Chown Command Changing the Owner of a File-The Chgrp Command Changing the Group of a File.	
3.0		**Shell-Command**	**6**
3	3.1	Using the Shell-Command Line Structure-Met characters-Creating New Commands-Command Arguments	
	3.2	Parameters-Program Output as Arguments-Shell Variables- -More on I/O Redirection-Looping in Shell Programs.	
4.0		**Filters**	**6**
4	4.1	Filters-The Grep Family-Other Filters-The Stream Editor Sed-The AWK Pattern Scanning and	
	4.2	Processing Language-Good Files and Good Filters.	
5.0		**Shell Programming**	**6**
5	5.1	Shell Variables-The Export Command-The Profile File a Script Run During Starting-The First Shell Script-The read Command-Positional parameters-The $? Variable knowing the exit Status-More about the Set Command-The Exit Command-Branching Control Structures-Loop Control Structures-The Continue and Break Statement	
	5.2	The Expr Command: Performing Integer Arithmetic-Real Arithmetic in Shell Programs-The here Document(<<)-The Sleep Command-Debugging Scripts-The Script Command-The Eval Command-The Exec Command.	
	5.3	The Process-The Meaning-Parent and Child Processes-Types of Processes-More about Foreground and Background processes-Internal and External Commands-Process Creation-The Trap Command-The Stty Command-The Kill Command-Job Control	

Text Books

1. The Unix programming Environment by Brain W. Kernighan & Rob Pike, Pearson.
2. Introduction to Unix Shell Programming by M.G.Venkateshmurthy, Pearson..

References

1. Unix and shell programming by B.M. Harwani, OXFORD university press..

ONLINE REFERENCES:

1. https://www.geeksforgeeks.org/introduction-linux-shell-shell-scripting/
2. https://www.tutorialspoint.com/unix/shell_scripting.htm
3. https://www.shellscript.sh/first.html

Internal Assessment (T1, T2 and FET)

3. T2 should be based on First to Fourth modules, for 10 marks.
4. FET shall be assessed for 5 marks separately.

End Semester Examination

7. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
8. The duration of end semester examination shall be Two hours.
9. The students need to solve all questions.
10. Question No.1 will be compulsory and based on entire syllabus.
11. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM022	Python for Data Science	02	00	-	02	--	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM022	Python for Data Science	10	-	5	15	40%	35	40%	50

Course Description

This course includes examples of analytics in a wide variety of industries, and students will learn how you can use analytics in their career and life. One of the most important aspects of this course is that you, the student, are getting hands-on experience creating analytics models.

Pre-requisites: The Joy of Computing using Python.

Course Objectives:

- Learn the syntax and semantics of Python Programming Language.
- Apply various data types and control structure
- Understand class inheritance and polymorphism
- Understand the object-oriented program design and development

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Write**[2] Python functions to facilitate code reuse and manipulate strings**.**
- **CO2** **Illustrate**[3] the process of structuring the data using lists, tuples and dictionaries.
- **CO3** **Demonstrate**[2] the use of built-in functions to navigate the file system.
- **CO4** **Appraise**[3] the need for working on web scraping.

Course Contents

Module	Unit	Description	Hours
1.0		**Data Structures and OOP**	**6**
1	1.1	Python Program Execution Procedure – Statements – Expressions – Flow of Controls –Functions – Numeric Data Types – Sequences – Strings – Tuples – Lists – Dictionaries.	
	1.2	Class – Constructors – Object Creation – Inheritance – Overloading. Text Files and Binary Files – Reading and Writing.	
2.0		**Numpy and Pandas Packages**	**6**
2	2.1	NumPy ndarray - Vectorization Operation - Array Indexing and Slicing - Transposing Array and Swapping Axes - Saving and Loading Array - Universal Functions - Mathematical and Statistical Functions in Numpy .	
	2.2	Series and DataFrame data structures in pandas - Creation of Data Frames – Accessing the columns in a DataFrame - Accessing the rows in a DataFrame - Panda's Index Objects - Reindexing Series and DataFrames - Dropping entries from Series and Data Frames - Indexing, Selection and Filtering in Series and Data Frames - Arithmetic Operations between Data Frames and Series - Function Application and Mapping.	
3.0		**Data Wrangling**	**6**
3	3	Combining and Merging Data Sets – Reshaping and Pivoting – Data Transformation – String manipulations – Regular Expressions.	
4.0		**Data Aggregation and Group Operations**	**8**
4	4.1	GroupBy Mechanics – Data Aggregation – GroupWise Operations –.	
	4.2	Transformations – Pivot Tables – Cross Tabulations – Date and Time data types	
5.0		**Visualization in Python**	**6**
5	5.1	Matplotlib and Seaborn Packages – Plotting Graph - Controlling Graphs – Adding Text – More Graph Types – Getting and Setting Values – Patches.	

Text Books

1. Al Sweigart, “Automate the Boring Stuff with Python”, William Pollock, 2015.

References

1. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist”, 2nd Edition, Green Tea Press, 2015.
2. Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd.
3. Wesley J Chun, “Core Python Applications Programming”, 3rd Edition, Pearson Education India, 2015
4. Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, “Data Structures and Algorithms in Python”, 1st Edition, Wiley India Pvt Ltd

Internal Assessment (T1, T2 and FET)

5. T2 should be based on First to Fourth modules, for 10 marks.
6. FET shall be assessed for 5 marks separately.

End Semester Examination

12. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
13. The duration of end semester examination shall be Two hours.
14. The students need to solve all questions.
15. Question No.1 will be compulsory and based on entire syllabus.
16. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM032	Computer Networks and Internet Protocol	02	00	-	02	--	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM032	Computer Networks and Internet Protocol	10	-	5	15	40%	35	40%	50

Course Description

Computer Networks and Internet Protocol is an introductory course designed to provide students with a comprehensive understanding of computer networking concepts and Internet protocols. The course covers the fundamental principles, technologies, and protocols that govern the operation of computer networks and facilitate communication over the Internet.

Pre-requisites: Basics of computer application.

Course Objectives:

- Understand the Fundamentals of Computer Networks.
- Understand the purpose and operation of each layer in the OSI and TCP/IP models.
- Study key networking protocols such as IP, TCP, UDP, and ICMP, and their roles in data transmission.
- Identify common network security threats and vulnerabilities.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Defining**[2] the basics of networking, components and underlying technologies.
- **CO2** **Illustrate**[3] the various key protocols in OSI model and TCP/IP protocol suite and explain various application protocols.
- **CO3** **Examine**[3] various transport protocols and its performance enhancing mechanisms.
- **CO4** **Determine**[4] the shortest path for the network using various routing protocols.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction**	**6**
1	1.1	Network terminologies, Network Models, Protocol layers and their services, Connection Oriented and Connection less services, Physical Media.	
2.0		**The Application Layer**	**6**
2	2.1	Principles of Application-Layer Protocols, HTTP, File Transfer: FTP, DNS, Electronic Mail in the Internet	
3.0		**The Transport Layer**	**6**
3	3.1	Transport-Layer Services and Principles, Multiplexing and Demultiplexing Applications, UDP and TCP, Connection Establishment,	
	3.2	Transport Layer Protocols (go back N, stop and wait, selective repeat), Flow Control, TCP Congestion Control	
4.0		**The Network Layer**	**8**
4	4.1	Introduction and Network Service Model, IP: the Internet IP addressing,	
	4.2	Routing Principles, Protocol, Routing in the Internet,	
5.0		**The Link Layer and Local Area Networks**	**6**
5	5.1	The Data Link Layer: Introduction, Services, Error Detection and Correction, Multiple Access Protocols.	
	5.2	LANs, LAN Addresses and ARP, IEEE standards and Ethernet	

Text Books

1. James Kurose, Keith Ross,‖ Computer Networking: A Top-Down Approach Featuring the Internet —, Addison Wesley
2. Andrew S. Tanenbaum ,‖Computer Networks —, Prentice-Hall Publishers

References

1. Larry Peterson , Bruce Davie ,‖Computer Networks a Systems Approach —, Morgan Kaufmann
2. William Stallings ,‖Data and Computer Communications‖, Prentice Hall

Internal Assessment (T1, T2 and FET)

1. T2 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 7 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM062	Object Oriented Modelling and Design	03	-	-	03	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U14MM062	Object Oriented Modelling and Design	10	10	5	25	40%	50	40%	75

Course Description

This course provides second-year computer engineering students with a comprehensive understanding of object-oriented modelling and design principles. Students will learn how to apply these principles to analyze, design, and implement software systems using object-oriented methodologies and techniques.

Pre-requisites: - Basics of computer system and programming

Course Objectives

- To understanding key concepts such as classes, objects, encapsulation, inheritance, and polymorphism.
- To earning various UML diagrams including class diagrams, object diagrams, use case diagrams, sequence diagrams, and state diagrams
- To Understanding when and how to apply design patterns to solve recurring design problems effectively
- To introduce software tools and platforms for object-oriented modeling and design, such as UML modeling tools and IDEs

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Apply**[3] object oriented modelling and design concepts to solve given problem using Rational Rose design tool.
- **CO2** **Design**[4] different data flow diagrams for given problem statement by using Rational Rose design tool.
- **CO3** **Sketch**[3] System design for given real time system using designing object oriented concepts.

- **CO4** **Design**[4] different UML diagrams for given problem statement by using STAR IIML design tool.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction**	**6**
1	1.1	Modeling as a design technique, Objects, classes, links and associations, Generalization..	
	1.2	Inheritance, Aggregation, abstract classes, generalization as extension and restriction, and multiple inheritances.	
2.0		**Dynamic & Functional Modeling**	**6**
2	2.1	Events and states, operations, nested state diagrams, concurrency advanced dynamic modeling concepts, relation of object and dynamic models,	
	2.2	Data Flow Diagrams, relation of functional to object and dynamic models	
3.0		**Design Methodology**	**6**
3	3.1	OMT methodology, Impact of an object oriented approach, Analysis Overview, System design with examples, Object Design, combining the three models,	
	3.2	Designing Algorithms, design association and physical packaging, design Optimization, implementation of Controls, desi gr association.	
4.0		**Structural Modeling using UML**	**6**
4	4.1	Classes, Relationships, Common mechanisms, Diagrams, Class Diagrams, Interfaces, Types and Roles, Packages, instances and Object Diagram	
	4.2	Use cases, Use case diagram, Interaction Diagrams and Activity diagrams, Events and signals, State Machines, Process and Threads, Time & Space, State chart diagrams.	
5.0		**Architectural Modeling**	**6**

5	5.1	Components, Deployment, Collaboration,	
	5.2	Patterns& frameworks Component Diagrams, Deployment Diagrams.	

Text Books

1. Erich Gamma, Richard Helm, Ralph Johnson and john Vlissides: Design Patterns – Elements of Reusable Object-Oriented Software, Pearson Education,2007
2. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML,2nd Edition, Pearson Education,2005

References

1. Grady Booch et.al.: Object-Oriented Analysis and Design with Applications,3rd Edition, Pearson Education,2007.
2. Booch, Jacobson, Rambaugh : Object-Oriented Analysis and Design with Applications, 3rd edition, Pearson, Reprint 2013

Internal Assessment (T1, T2 and FET)

1. T1 should be based on first two modules and T2 should be based on next two modules, for 10 marks each.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 50 marks comprise of 5 questions, each carrying 10 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all 5 questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14OE021	Advance Microsoft Office	01	02	-	01	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
U14OE021	Advance Microsoft Office	50	40%	-	-	50

Course Description

This course is at even semester of first year Technology. Microsoft office is a powerful suite of tools that can help professionals stay organized, collaborate with colleagues, and create professional documents. From creating reports and proposals to analysing data in Excel to managing email and scheduling in Outlook, Microsoft has something for every professional.it

Helps to learn Microsoft word from beginner to expert level. Learn to create a professional document template

Pre-requisites: Computer Basics

Course Objectives

- To know how to use the most common Microsoft office programs..
- To be able to create documents for printing and sharing.
- To be able to create and share presentations.
- Gaining hands-on experience in becoming a user of Microsoft office

Course Outcomes: After the end of this course students will be able to

- **CO1** **Creating**[6] and formatting simple documents**.**
- **CO2** **Creating**[6] simple reports using charts and graphs.
- **CO3** **Creating**[6] dynamic and engaging presentations with animations and transitions.
- **CO4** **Create**[6] a simple report for a given technical topic, and typeset it in Microsoft office

Course Contents

Module	Unit	Description	Hours
1.0		**Word Processing Package**	**7**
1	1.1	Introduction-Features- Word User Interface Elements- Creating New Documents- Basic Editing-Saving a Document-Printing a Document- Print Preview-Page Orientation Viewing Documents-Setting. Tabs-Page Margins- Indents- Ruler-Formatting Techniques Font Formatting-Paragraph Formatting- Page Setup-Headers &Footers-Bullet.	
	1.2	Numbered List Borders and Shading- Find and Replace-Page Break Page Numbers-Mail Merging-Spelling and Grammar Checking- Thesaurus- Macros-Tables- Side By-Side and Nested Tables- Formatting Tables- Drawing- Word art- Paint Brush	
2.0		**Spreadsheet Package**	**8**
2	2.1	Introduction-Excel User Inter face Working With Cell and Cell Addresses-Selecting a Range, Moving, Cutting, Copying With Paste-Inserting and Deleting Workbook- Cells- Freezing Renaming Cells- a Worksheet- Adding, Deleting Cell, Contents Height Selecting Using and Copying Borders- / Width-Formatting in Print a Worksheet Cell- Boxes Area-Margin Using and Colours- Within Print a Worksheet and Preview- Fonts-	
	2.2	Cantering Orientation- Automatically- Aligning-Wrapping Preparing a Heading, Cantering Worksheet Insert a and Worksheet- Comments- for Rotating Row/ the Column Printer-Using Clear Text Formatting Options- Formatting Changing Header and Footer- Inserting Page Breaks- Sorting Data	
3.0		**Advanced Features of Excel**	**8**
3	3.1	All Functions in Excel- Using Logical Functions Statistical Functions-Mathematical Functions – Linking Data between Worksheet Elements of Excel Charts-Categories.	
	3.2	Create a Chart- Choosing Chart Type- Edit Chart Axis –Titles, Labels, Data Series and Legend- Adding a Text Box- Rotate Text in a Chart.	
4.0		**Presentation Package**	**7**
4	4.1	Advantages of Presentation-Screen Layout- Creating Presentation- Inserting Slides-Adding Sounds and Videos-Formatting Slides -Slide Layout Views in Presentation	
	4.2	Colours Scheme-Background Action Buttons- Slide Transition- Custom Animation- Managing Slide Shows – Using Pen Setting Slide Intervals	

Suggested List of Experiments

1. Create a new document, explore different templates, and understand the document structure.
2. Practice mail merging using a sample data source.
3. Record and use a simple macro to automate a repetitive task.
4. Insert, delete, and rename worksheets; freeze panes; and work with workbook elements.
5. Explore advanced formatting options, page layout settings, and print a worksheet. Sort and filter data in a worksheet
6. Use logical, statistical, and mathematical functions in Excel Link data between worksheets, create different types of charts, and customize chart elements.
7. Create a new presentation, explore different layouts, and add slides, Apply a color scheme, background, and formatting options to slides
8. Insert sounds and videos into slides, Apply slide transitions and custom animations to enhance presentation flow, Practice managing slide shows, including pen settings and slide intervals.

Text Books

1. Microsoft Office 365 & Office 2019: Advanced" by Misty E. Vermaat
2. Microsoft Office 2019 Inside Out" by Joe Habraken,

References

1. Excel 2019 Bible" by Michael Alexander and Richard Kusleika
2. Power Excel with MrExcel: Master Pivot Tables, Subtotals, Visualizations, VLOOKUP, Power BI, and Data Analysis" by Bill Jelen

Evaluation Scheme

4. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.
5. The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.
6. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U01EM002	Engineering Management	02	-	-	02	-	-

Evaluation Scheme

Course Code	Course Name	Evaluation Scheme (In Semester)					End Semester Exam (ESE)		
		T1	T2	FET	Total	Min pass	Marks	Min pass	Total (Marks)
U01EM002	Engineering Management	-	10	5	15	40%	35	40%	50

Course Description

This course presents the principles and techniques of managing engineering projects from the initiation phase, through planning, execution, control and closeout. Students will develop the analytical skills and awareness necessary on the management side of engineering projects.

Pre-requisites: -

Course Objectives

- Understand the importance of management in engineering field.
- Apply knowledge of project management
- Understand the concept of material management.
- Explain importance of quality management
- **Course Outcomes:** After the successful completion of the course students will able to:

- **CO1** **Explain**[2] function of principles of management.
- **CO2** **Correlate**[3] knowledge of project management to various engineering project
- **CO3** **Apply**[3] knowledge of material management
- **CO4** **Explain2** importance of quality management in engineering field.

Course Contents

Module	Unit	Description	Hours
1.0		**Introduction To Management**	**06**
1	1.1	Principles of Management (Henry Fayol) Functions of Management: Planning- Organizing ,Staffing Directing Supervision, Co-ordination, Communication, Motivation, Leading Controlling –Decision Making: Process, introduction to decision tree	
2.0		**Project Management**	**06**
2	2.1	Introduction, Need for Project Management, , The Project Life Cycle, Phases of Project Management, need of project management , Role of Project Manager (PM), Work Breakdown Structure (WBS)	
	2.2	Project Planning- Time Management, Tools for time management Bar Chart/ Gantt Charts, Mile stone chart- representation, uses, progress monitoring.	
3.0		**Project Control-**	**06**
3	3.1	Network analysis- Basic definition application, rules for drawing networks Critical Path Method (CPM)-Development of CPM Network Time Estimates, Calculations of Floats, Critical Path.	
	3.2	Programme Evaluation Review Techniques (PERT)- Time Estimates, Slack calculation , Probability of Project Completion.	
4.0		**Project Monitoring**	**06**
4	4.1	Project Network Updating, - introduction, process, data required for updating, steps in updating application	

	4.2	Cost Control- introduction, project cost. Crashing/compression of network. Application	
5.0		**Material Management and Quality Control**	**06**
5	5.1	Objectives, Need for Inventory Control, EOQ Analysis, ABC analysis, Safety Stock, Purchase Procedure, Stores Record , site layout	
	5.2	Quality Management: Importance, Quality Circle, ISO 9000, QA, QC, QMS –Purpose	

References

1 Chitkara, K.K. Construction Project Management: Planning, Scheduling and Control, McGraw Hill Publishing Company, New Delhi, 1998.

2 Jha, Kumar Neeraj., Construction Project management, Theory & Practice, Pearson Education India, 2015.

3 Punmia, B.C., Khandelwal, K.K., Project Planning with PERT and CPM, Laxmi Publication

Internal Assessment (T1, T2 and FET)

1. T2 should be based on First to Fourth modules, for 10 marks.
2. FET shall be assessed for 5 marks separately.

End Semester Examination

1. Question paper will be of 35 marks comprise of 5 questions, each carrying 07 marks
2. The duration of end semester examination shall be Two hours.
3. The students need to solve all questions.
4. Question No.1 will be compulsory and based on entire syllabus.
5. Remaining question (Q.2 to Q.5) will be selected from all the modules

****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14VS401	Software Proficiency I (Java Programming)	-	04	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term work	Min pass	Marks	Min pass	Total (Marks)
U14VS401	Software Proficiency I (Java Programming)	25	40%	25	40%	50

Course Description

This course is designed to provide second-year computer engineering students with a solid foundation in Java programming language. Students will learn the fundamentals of Java programming, object-oriented principles, and practical application development techniques.

Pre-requisites: - U14PC303- Object Oriented Programming using C++

Course Objectives

- To learn basic principles of programming languages and programming paradigms
- To learn structuring the data and manipulation of data, computation and program structure
- To learn Object Oriented Programming (OOP) principles using Java Programming Language
- To learn basic concepts of logical and functional programming language.
- To Implement efficient and scalable Java applications, considering best practices and design patterns.

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Apply**[3] various object oriented features like inheritance, data abstraction, encapsulation and polymorphism to solve mathematical problems using JDK
- **CO2** **Apply**[3] the concept of multithreading, I/O operations, exception and networking to execute and handle multiple programs by using JDK
- **CO3** **Develop**[4] GUI application with database connectivity by using the concept of Swing and Applet.
- **CO4** **Evaluate**[5] the compile time and run time error by using appropriate syntax.
- **CO5** **Develop**[4] application to solve real world problems by using java language

Course Contents

Module	Unit	Description	Hours
1.0		**Fundamentals of Programming**	**7**
1	1.1	Programming paradigms- Introduction to programming paradigms, Introduction to four main Programming paradigms- procedural, object oriented, functional, and logic & rule based.	
	1.2	Elementary Data Types : Primitive data Types, Character String types, User Defined Ordinal Types, Array types, Associative Arrays, Record Types, Union Types, Pointer and reference Type. Expression and Assignment Statements: Arithmetic expression, Overloaded Operators, Type conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed mode Assignment. Statement level Control Statements: Selection Statements, Iterative Statements, And Unconditional Branching. Abstract Data Types and Encapsulation Construct: Design issues for Abstraction, Parameterized Abstract Data types, Encapsulation Constructs, Naming Encapsulations	
2.0		**Fundamentals of JAVA, Arrays**	**8**
2	2.1	One dimensional array, multi-dimensional array, alternative array declaration statements, String Handling: String class methods, Managing I/O: Streams, Byte Streams and Character Streams, Predefined Streams, Reading Console Input, Writing Console Output, Print Writer class.	
	2.2	Classes and Methods: class fundamentals, declaring objects, assigning object reference variables, adding methods to a class, returning a value, constructors, this keyword, garbage collection, finalize() method, overloading methods, argument passing, object as parameter, returning objects, access control, static, final, nested and inner classes, command line arguments, variable -length arguments.	
3.0		**Inheritance, Packages and Exception Handling**	**8**
3	3.1	Inheritances: member access and inheritance, super class references, Using super, multilevel hierarchy, constructor call sequence, method overriding, dynamic method dispatch, abstract classes, Object class. Packages and Interfaces: defining a package, finding packages and CLASSPATH, access protection, importing packages, interfaces (defining, implementation, nesting, applying), variables in interfaces, extending interfaces, instance of operator.	
	3.2	Fundamental of exception, exception types, uncaught exceptions, try, catch, throw, and throws, finally, multiple catch clauses, nested try statements, built-in exceptions, and custom exceptions (creating your own exception sub classes).	
4.0		**Presentation Package**	**7**
4	4.1	Concurrency and Synchronization, Java Thread Model: Thread priorities, Synchronization, Messaging, Main Thread, Creating thread: Implementing Thread using thread class and Runnable interface. Creating multiple threads using isAlive() and join()	

	4.2	Web Based Application in Java: Use of JavaScript for creating web based applications in java, Introduction to javascript frameworks- React, Vue, Angular	
5.0		**GUI ad Database Connectivity**	
5	5.1	Introduction to AWT programming Layout and component managers Event handling Applet class Applet life-cycle Passing parameters embedding in HTML	**7**
	5.2	Swing components – JApplet, JButton, JFrame, etc. Sample swing programs	

Suggested List of Experiments

All the experiments will be based on the course content

1. Programs using constructor and destructor.
2. Creation of classes and use of different types of functions.
3. Count the number of objects created for a class using static member function.
4. Write programs on interfaces.
5. Write programs on packages.
6. Write programs using function overloading.
7. Programs using inheritance.
8. Programs using IO streams.
9. Programs using files.
10. Write a program using exception handling mechanism.
11. Programs using AWT
12. Programs on swing.
13. Programs using JDBC.

References

1. Herbert Schildt, "The Complete Reference Java", 9th Ed, TMH,.
2. Dr.R. Nageshwar Rao, "Core Java: An Integrated Approach", Dreamtech Press.
3. Programming with Java A Primer, E. Balaguruswamy Tata McGraw Hill Companies
4. Java Programming John P. Flynt Thomson 2nd.

Evaluation Scheme

1. TERM WORK assessment shall be based on the overall performance of the student with every assignment graded from time to time.
2. The grades will be converted to marks as per ‘credit and grading system’ manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14PC404	Data Structure Lab	-	02	-	-	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
U14PC404	Data Structure Lab	-	-	25	40%	25

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Apply**[3] operations of linear data structure to solve the given problem using a programming language.
- **CO2** **Apply**[3] operations of nonlinear data structure to solve the given problem using a programming language
.
- **CO3** **Analyze**[4] searching and sorting techniques for solving search and sort problems using asymptotic notation
.

Suggested List of Experiments

All the experiments will be based on the course content of **Data Structure - U14PC401**

Programs based on

1. array, function, pointer, structures
2. Singly Linked List
3. Doubly Linked List
4. Circular Linked List
5. Stack ADT – Static and Dynamic
6. Queue ADT – Static and Dynamic
7. Stack application, circular and double ended queue
8. Searching – Linear, Binary and Hashing
9. Sorting – Bubble, Selection, Insertion,
10. Sorting – Merge and Quick
11. Binary Search Tree, Traversal of Trees

12. Graph using adjacency list and traversal

References

1. Data Structure using C A. M. Tanenbaum,Y. Langsam, M.J. Augenstein Prentice- Hall Of India Pvt. Limited 2003
2. Understanding Pointers in C Yashavant Kanetkar BPB Publication I 2009

Evaluation Scheme:

1. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.
2. The grades will be converted to marks as per 'credit and grading system' manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14PC405	Database Management System Laboratory	-	02	-	-	02	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
U14PC405	Database Management System Laboratory	20	40%	-	-	25

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Design**[4] an ER diagram and convert it into relational database to solve the given problem through applying basic concepts database engineering
- **CO2** **Apply**[3] DDL and DML SQL queries to solve the given problems using a database management system
- **CO3** **Create**[6] and Exhibit solution for defined real world problem using hashing techniques, transaction processing, concurrency control etc.

List of Experiments

All the experiments will be based on the course content of **Database Management System- U14PC403**

1. Drawing an E-R Diagram for any organization. Converting E-R diagram into Relational Tables.
2. Installation and Demonstration of DBMS Oracle / MySQL / SQL Server / PostgreSQL etc.
3. Implementation of Data Definition Language (DDL) Queries (e.g. create, alter and drop tables).
4. Implementation of Data Manipulation Language (DML) Queries (e.g. insert, delete, update and select statements).
5. Implementation of Basic SQL SELECT statement for displaying / extracting data from single table or multiple tables.
6. Implementation of SQL constructs for aggregating data, use of group by, having clauses.
7. Implementation of nested sub-queries, complex queries, views and Joins.
8. Implementation of Triggers.
9. Implementation of Functions and Stored Procedures.
10. Implementation of Database connectivity with object oriented language (Java).

References

1. Fundamentals of Database Systems Ramez Elmasri and Shamkant Navathe Pearson Education 4 edition 2007.
2. Systems: Design, Implementation and management Peter Rof, Carlos Coronel Cengage Learning 7 edition 2014.
3. Principles of Database Systems J. D. Ullman Galgotia publications 1 edition 2011
4. SQL: A Complete Reference Alexis Leon, Mathews Leon McGraw Hill Education 1 edition 2002

Evaluation Scheme

1. TERM WORK assessment must be based on the overall performance of the student with every assignment done on computer graded from time to time.
2. The grades will be converted to marks as per 'credit and grading system' manual and should be added and averaged.
3. Based on above scheme grading and TERM WORK assessment should be done.

*****

Course Code	Course Name	Teaching Scheme (Hr/week)			Credits Assigned		
		Theory	Practical	Tutorial	Theory	Practical	Tutorial
U14MM066	Object Oriented Modelling and Design Lab	-	02	-	-	01	-

Evaluation Scheme

Course Code	Course Name	In Semester Evaluation		End Semester Exam (OE/POE)		
		Term Work	Min pass	Marks	Min pass	Total (Marks)
U14MM066	Object Oriented Modelling and Design Lab	-	-	25	40%	25

Course Outcomes: After the successful completion of the course students will able to:

- **CO1** **Apply**[3] object oriented modelling and design concepts to solve given problem using Rational Rose design tool.
- **CO2** **Sketch**[3] System design for given real time system using designing object oriented Concepts.
- **CO3** **Design** [4] different UML diagrams for given problem statement hy using STAR IIML Design tool.

.

Suggested List of Experiments

All the experiments will be based on the course content of **Object Oriented Modelling and Design - U14MM062**

Programs based on

1. Introduction of object oriented analysis and object oriented design.
2. Object Oriented Modeling, Choose a hypothetical system of significant complexity (on your project topic) and write an SRS.
3. Draw one or more Use Case diagrams for capturing and representing requirements of the system. Use case diagrams must include template showing description and steps of the Use Case for various scenarios.
4. Draw basic class diagrams to identify and describe key concepts like classes, types in your system and their relationships.
5. Draw sequence diagrams OR communication diagrams with advanced notation for

your system to show objects and their message exchanges.

6. Draw activity diagrams to display either business flows or like flow charts.
7. Draw component diagrams assuming that you will build your system reusing existing components along with a few new ones.
8. Draw deployment diagrams to model the runtime architecture of your system.

Text Books

1. Erich Gamma, Richard Helm, Ralph Johnson and john Vlissides: Design Patterns – Elements of Reusable Object-Oriented Software, Pearson Education,2007
2. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML,2nd Edition, Pearson Education,2005

References

1. Grady Booch et.al.: Object-Oriented Analysis and Design with Applications,3rd Edition, Pearson Education,2007.
2. Booch, Jacobson, Rambaugh : Object-Oriented Analysis and Design with Applications, 3rd edition, Pearson, Reprint 2013

Evaluation Scheme:

4. TERM WORK assessment must be based on the overall performance of the student with every assignment graded from time to time.
5. The grades will be converted to marks as per 'credit and grading system' manual and should be added and averaged.
6. Based on above scheme grading and TERM WORK assessment should be done.

*****