

F.Y.I.T

Semester-I	
Programming Principles with C	
Course Code: USIT101	Course Objectives: <ol style="list-style-type: none"> 1. To develop the logical ability of the student. 2. Basic concepts to be cleared using suitable examples. 3. Different approach towards the problem. 4. To handle the errors and find suitable solution. 5. Debugging the code.
	Course Outcomes: Learners will be able to, <ol style="list-style-type: none"> 1. Learn the basic principles of programming. 2. Develop of logic using algorithm and flowchart. 3. Acquire the information about data types. 4. Understanding of input and output functions. 5. Enhance advanced concepts using program.
Programming Principles with C Practical	
Course Code: USIT1P1	Course Objectives: <ol style="list-style-type: none"> 1. To develop the logic of the student. 2. Describe loops and decision making using programs. 3. Practical use of operators. 4. Illustration of the difficult concepts using programming examples. 5. Discussion of the relevant concepts using program.
	Course Outcomes: Learners will be able to, <ol style="list-style-type: none"> 1. Develop applications. 2. Work with textual information, characters and strings. 3. Understand of a functional hierarchical code organization 4. Debug the program 5. Understand the differences between syntax errors, runtime errors, and logic errors.
Digital Logic and Applications	
Course Code: USIT102	Course Objectives: <ol style="list-style-type: none"> 1. To introduce the basics of logic in digital electronics as an entry level course. 2. To interpret and assess number systems and the conversions of number systems 3. To analyze the boolean expressions and reduce the expression to the minimum. 4. To design simple logic circuits using tools such as Boolean Algebra and Karnaugh Mapping. 5. To understand the state of a memory cell and its types using flip-flops. 6. To create simple digital systems using counters, registers etc.
	Course Outcomes: Learners will be able to, <ol style="list-style-type: none"> 1. Apply number conversion techniques in real digital systems 2. Solve boolean algebra expressions

	<p>3. Derive and design logic circuits by applying minimization in SOP and POS forms</p> <p>4. Design and develop Combinational and Sequential circuits</p> <p>5. Understand and develop digital applications</p>
Digital Logic and Applications Practical	
Course Code: USIT1P2	<p>Course Objectives:</p> <ol style="list-style-type: none"> 1. To apply and test the gates learnt using various IC's . 2. To evaluate the Boolean expression to reduce and minimize the gates used
	<p>Course Outcomes: Learners will be able to,</p> <ol style="list-style-type: none"> 1. Construct basic and universal logic circuits. 2. Verify the functionalities of various IC's. 3. Design circuits using K-maps minimization technique 4. Design and test Encoders, Decoders, Multiplexers and Demultiplexers 5. Design and develop logic for Registers, Counters and its applications.
Fundamentals of Database Management Systems	
Course Code: USIT103	<p>Course Objectives: The objective of the course is to present an introduction to fundamentals of database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information from a DBMS.</p>
	<p>Course Outcomes: Learners will be able to</p> <ol style="list-style-type: none"> 1. Define and describe the fundamental elements of relational database management system. 2. To relate the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL. 3. Design ER-models to represent simple database application scenarios. 4. Transform the ER-model to relational tables, populate relational database and formulate SQL queries on data. 5. Improve the database design by normalization. 6. Understand basic database storage structures and access techniques: file and page organizations, indexing methods and hashing.
Fundamentals of Database Management Systems Practical	
Course Code: USIT1P3	<p>Course Objectives:</p> <ol style="list-style-type: none"> 1.To introduce ER data model, database design and normalization. 2 .To Learn SQL basics for data definition and data manipulation.
	<p>Course Outcomes: Learners will be able to:</p> <ol style="list-style-type: none"> 1. Design database schema for a given application and apply normalization. 2. Acquire skills in using SQL Commands for data Definition and data manipulation.
Computational Logic and Discrete Structures	
Course Code: USIT104	<p>Course Objectives:</p> <ol style="list-style-type: none"> 1.Course will provide students with an overview of discrete mathematics.

	2.Students will learn about topics such as logic and proofs, sets and functions, recursion, graph theory, trees and other important discrete math concepts.
	Course Outcomes: Learners will be able to: 1. Use logical notation 2. Perform logical proofs 3. Apply recursive functions and solve recurrence relations 4. Use graphs and trees 5. Apply basic and advanced principles of counting 6. Define sets and Relations 7. Calculate discrete probabilities.
Computational Logic and Discrete Structures Practical	
Course Code: USIT1P4	Course Objectives: Course will make students understand different commands and functions of SCILAB. It will enable student to use these tools to compute solutions of various discrete mathematical structures.
	Course Outcomes: Learners will be able to: 1. To find computational solution to various discrete mathematical structures.
Technical Communication Skills	
Course Code: USIT105	Course Objectives: 1. To recognize the importance of various types of communication in technical set up. 2.To understand the dynamics in different forms of formal communication. 3. To learn about active listening and the art of giving presentations and interviews. 4.To learn the art of business writing and ethics in business communication across functional areas. 5.To evaluate, analyze and interpret technical data.
	Course Outcome: Learners will be able to, 1. Analyze, synthesize and utilize the process and strategies from delivery to solving communication problem. 2. Learn the communication methodologies at workplace and learning about importance of team collaboration. 3. Learn about different technical communication such as presentations and interviews. 4. Understand and apply the art of written communication in writing reports, proposals. 5. Ground rules of ethical communication and MIS. 6. Understand the functions of graphs, maps, charts.
Technical Communication Skills Practical	
Course Code: USIT1P5	Course Objectives: 1.To express thoughts feelings and ideas of learners by using features of MS Word. 2.To articulate formal and informal reports. 3.To analyze and interpret data and learn visualization of data. 4.To learn effective tools of presentation.
	Course Outcome: Learners will be able to:

	<ol style="list-style-type: none"> 1. Use different forms of digital mediums for effective communication. 2. Create technical documents and format existing documents for effective communication. 3. Learn to use graphical tools for better visualization. 4. Create business presentation effectively. 5. Visualize the data from pictorial representations.
Semester-II	
Object Oriented Programming with C++	
Course Code: USIT201	Course Objectives: <ol style="list-style-type: none"> 1. Understand object oriented programming and advanced C++ concepts 2. Be able to explain the difference between object oriented programming and procedural programming. 3. Be able to program using more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc. 4. Be able to build C++ classes using appropriate encapsulation and design principles 5. Be able to apply object oriented or non-object oriented techniques to solve 6. bigger computing problems
	Course Outcomes: Learners will be able to, <ol style="list-style-type: none"> 1. Understand the concept of OOPs, feature of C++ language. 2. Understand and apply various types of Datatypes, Operators, Conversions while designing the program. 3. Understand and apply the concepts of Classes & Objects, friend function, constructors & destructors in program design. 4. Design & implement various forms of inheritance, String class, calling base class constructors. 5. Apply & Analyze operator overloading, runtime polymorphism, Generic Programming. 6. Analyze and explore various Stream classes, I/O operations and exception handling.
Object Oriented Programming with C++ Practical	
Course Code: USIT2P1	Course Objectives: <ol style="list-style-type: none"> 1. The student should be able to explain the important characteristics of the C++ programming language. 2. The learner must be able to combine components of the C++ programming language to develop structured program. 3. The student must demonstrate the skills essential to compile, debug, and test C++ programs correctly.
	Course Outcomes: Learners will be able to, <ol style="list-style-type: none"> 1. Utilize C++ characteristics in software design and development. 2. Explain object-oriented techniques and explain how C++ supports them. 3. Employ C++ to demonstrate practical skill developing object-oriented solutions. 4. Examine a problem statements and design and develop object-oriented software using good coding practices and procedures.

	5. In object-oriented design, use common software patterns and recognize their relevance in other software development contexts.
Fundamentals of Micro Processor and Microcontrollers	
Course Code: USIT202	Course Objectives: 1. To understand the basic concept of Micro Computer Systems 2. To develop background knowledge in 8085 Microprocessor 3. To write Assembly language Programs of 8085 4. To understand the peripheral devices and interfacing to 8051 Micro Controller and design aspects of Micro Controller
	Course Outcomes: Learners will be able to, 1. Understand the basic concepts of Micro Computer Systems 2. Understand the architecture and hardware aspects of 8085 3. Write assembly language programs in 8085 4. Design elementary aspects of Micro Controller based systems 5. Interfacing peripherals using Micro Controller
Fundamentals of Micro Processor and Microcontrollers Practical	
Course Code: USIT2P2	Course Objectives: The course aims to learn 1. Operations related to single & Multiple memory locations 2. Simple assembly language programs 3. How to perform register operations, packing and unpacking 4. Embedding computer using 8051 microcontrollers 5. Interfacing I/O Ports 6. Implement flashmagic in microcontrollers
	Course Outcomes: Learners will be able to, 1. Apply concepts of 8085 to single & Multiple Memory Locations 2. Apply concepts of micro-processor register operations 3. Can implement assembly language programs 4. Use of Shift registers 8 & 16 bits 5. Apply the knowledge of Flash Magic in embedded Controllers 6. Learns to simulate and configure different timer controls
Web Applications Development	
Course Code: USIT203	Course Objectives: 1. Understand basic concepts of Internet and World Wide Web. 2. Comprehend different HTML elements that can be used to develop static web pages. 3. Become familiar with concept of stylesheets and various CSS effects. 4. Peruse JavaScript as a tool to add dynamism to static HTML pages. 5. Explore how server-side script works on the web. 6. Learn how PHP can be connected to a database to store and retrieve data.
	Course Outcomes: 1. Analyze working of Internet. 2. Gain an insight into designing web pages. 3. Use different ways of styling web pages using CSS. 4. Implement basic and complex functionalities of JavaScript in a web page. 5. Employ PHP Scripts to execute dynamic tasks in a web page. 6. Perform various database tasks using PHP.
Web Application Development Practical	

Course Code: USIT2P3	Course Objectives: 1. Understand how to effectively implement HTML. 2. Write CSS effectively to create well organized, styled web pages. 3. Add versatility to a web page with client-side scripting. 4. Deploy a local web server and run a simple web application. 5. Read and process data in MySQL using PHP.
	Course Outcomes: Learners will be able to, 1. Design static web pages using Hyper Text Markup Language (HTML). 2. Enhance the look of web pages by implementing CSS. 3. Collect information from the user with HTML Forms. 4. Design interactive webpages using client-side script (JavaScript). 5. Implement Document Object Model and events in web pages using JavaScript. 6. Write and deploy basic PHP code to simplify web development. 7. Store and retrieve data from a server using PHP.
Numerical Methods	
Course Code: USIT204	Course Objectives: Course will enhance the problem solving skills of students using extremely powerful numerical methods
	Course Outcomes: Learners will be able to, 1. Understand numerical techniques to find the roots of non-linear equations and solution of system of linear equations. 2. Understand the difference operators and the use of interpolation. 3. Understand numerical differentiation and integration and numerical solutions of ordinary and partial differential equations.
Numerical Methods Practical	
Course Code: USIT2P4	Course Objectives: Course will provide different tools to find solutions to various numerical techniques
	Course Outcomes: Learners will be able to, 1. Find fast and accurate solution to simple and complex numerical problems using these programs
Green IT	
Course Code: USIT205	Course Objectives: 1. To understand the concept of Green Technology. 2. To learn Green IT regulating Green IT and different standards. 3. To understand the concept of minimizing power utilization in technology. 4. To know about Green PCs, Green notebooks and servers and Green data centers. 5. To know how the way of work is changing and understand implementation of Paperless work. 6. To know the concept of Recycling. 7. To understand Metrics for Green IT.
	Course Outcomes: Learners will be able to, 1. Understand the concept of Green IT and problems related to it. 2. Know different standards for Green IT. 3. Understand the how power usage can be minimized in Technology. 4. Learn about how the way of work is changing. 5. Understand the concept of recycling.

	6. Know how information system can stay Green Information system.
Practical's in PL/SQL	
Course Code: USIT2P5	Course Objectives: <ol style="list-style-type: none"> 1. To understand the basics of PL/SQL. 2. To understand control and conditional statement in PL/SQL. 3. To understand working of sequences and cursor in PL/SQL. 4. To understand concept of stored procedure and functions. 5. To understand triggers and packages in PL/SQL. 6. To understand the concept of Exception handling.
	Course Outcomes: Learner will be able to: <ol style="list-style-type: none"> 1. Understand the basics of PL/SQL. 2. Use of the control and conditional statement in PL/SQL. 3. Apply sequences and cursor in PL/SQL. 4. Know the concept of stored procedure and functions 5. Create the triggers and packages in PL/SQL. 6. Implement the concept of Exception handling.

S.Y.I.T.

Semester-III	
Python Programming	
Course Code: USIT301	<p>Course Objective:</p> <ol style="list-style-type: none"> 1. Interpret the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements. 2. Express proficiency in the handling of strings and functions. 3. Determine the methods to create and manipulate Python programs by utilizing the data structures like lists, dictionaries, tuples and sets. 4. Identify the commonly used operations involving file systems and regular expressions. 5. Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance and polymorphism as used in Python.
	<p>Course Outcome: After completing the course, the learner will be able to:</p> <p>CO1: Aware of the variables, expressions, looping and conditions used in Python programming.</p> <p>CO2: Implement functions, strings, lists, tuples and directories</p> <p>CO3: Create GUI forms and add widgets.</p> <p>CO4: Use MySQL to store data.</p> <p>CO5: Apply the programming skillset learnt here into various domains by having advance programming skillset of Python and usage of libraries.</p>
Data Structures	
USIT302	<p>Course Objective:</p> <ol style="list-style-type: none"> 1. Ability to analyze the performance of algorithms. 2. Ability to choose appropriate algorithm design techniques for solving problems. 3. Understand how the choice of data structures and the algorithm design methods impact the performance of programs.
	<p>Course Outcome: After completing the course, the learner will be able to:</p> <p>CO1: Identify and distinguish data structure classification, data types, their complexities</p> <p>CO2: Implement array, linked list, stack and queue.</p> <p>CO3: Implement trees, various hashing techniques and graph for various applications</p> <p>CO4: Compare various sorting and searching techniques</p>
Computer Networks	
Course Code: USIT303	<p>Course Objective:</p> <ol style="list-style-type: none"> 1. Knowledge of uses and services of Computer Network. 2. Ability to identify types and topologies of network. 3. Understanding of analog and digital transmission of data. 4. Familiarization with the techniques of routing. 5. Understand the functioning of networking application
	<p>Course Outcomes: After completing the course, the learner will be able to:</p> <p>CO1: Identify various data communication standards, topologies and terminologies</p>

	<p>CO2: Describe how signals are used to transfer data and communication aspects between nodes</p> <p>CO3: Configure IP addresses using TCP/IP protocol suite</p> <p>CO4: Use different application layer protocols</p>
Operating Systems	
Course Code: USIT304	<p>Course Objective:</p> <ol style="list-style-type: none"> 1. Analyze the concepts of processes in operating system and illustration of the scheduling of processor for a given problem instance. 2. Identify the dead lock situation and provide appropriate solution so that protection and security of the operating system is also maintained. 3. Analyze memory management techniques, concepts of virtual memory and disk scheduling. 4. Understand the implementation of file systems and directories along with the interfacing of IO devices with the operating system. 5. Ability to apply CPU scheduling algorithms to manage tasks. 6. Initiation into the process of applying memory management methods and allocation policies. 7. Knowledge of methods of prevention and recovery from a system deadlock.
	<p>Course Outcomes: After completing the course, the learner will be able to:</p> <p>CO1: Role of Operating System Computer System.</p> <p>CO2: Use the different types of Operating System and their services.</p> <p>CO3: configure process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.</p> <p>CO4: Apply virtual memory concepts.</p> <p>CO5: Effectively use and manage secondary memory.</p>
Applied Mathematics	
Course Code: USIT305	<p>Course Objective: The course is aimed to develop the basic Mathematical skills of IT students that are imperative for effective understanding of IT subjects.</p> <ol style="list-style-type: none"> 1. Apply the knowledge of matrices to solve the problems. 2. Know and to understand various types of numerical methods. 3. Ability to interpret the mathematical results in physical or practical terms for complex numbers. 4. Inculcate the habit of Mathematical Thinking through Indeterminate forms and Taylor series expansion 5. solve and analyze the Partial derivatives and its application in related field of engineering
	<p>Course Outcomes: Upon the successful completion of the course, students will be able to:</p> <p>CO 1: Solve the matrix operations, identify the linear dependence and independence of a vectors. 17</p> <p>CO 2: Familiar with the various forms and operations of a complex number.</p> <p>CO 3: Find the Laplace transform of a function and Inverse Laplace transform of a function using definition also solve ordinary differential equations using Laplace transform.</p>

	<p>CO 4: Evaluate the multiple integrals in Cartesian, Polar coordinates, change the order of the integral,</p> <p>CO 5: Apply integration methods to calculate the areas and volumes of solids.</p> <p>CO 6: Evaluate the Beta, Gamma, Differentiation Under integral sign and error functions</p>
Semester – IV	
Java Programming	
Course Code: USIT401	<p>Course Objectives: Upon completion of this course, students will be able to:</p> <ol style="list-style-type: none"> 1. Understand the concept of OOP as well as the purpose and usage principles of inheritance, polymorphism, encapsulation and method overloading. 2. Identify classes, objects, members of a class and the relationships among them needed for a specific problem. 3. Create Java application programs using sound OOP practices (e.g., interfaces and APIs) and proper program structuring (e.g., by using access control identifies, automatic documentation through comments, error exception handling). 4. Use testing and debugging tools to automatically discover errors of Java programs as well as use versioning tools for collaborative programming/editing. 5. Develop programs using the Java Collection API as well as the Java standard class library. 6. Apply object-oriented programming concepts in problem solving through JAVA.
	<p>Course Outcome:</p> <p>After completing the course, the learner will be able to:</p> <p>CO1: Learn the architecture of Java</p> <p>CO2: Identify data types, control flow, classes, inheritance, exceptions and event handling</p> <p>CO3: Use object-oriented concepts for problem solving real-life applications</p> <p>CO4: Build GUI programs</p> <p>CO5 : Create event driven programs using java.</p>
Introduction to Embedded Systems	
Course Code: USIT402	<p>Course Objectives:</p> <ol style="list-style-type: none"> 1. To introduce the Building Blocks of Embedded System 2. To Educate in Various microcontrollers used in Embedded Development 3. To Introduce Bus Communication in processors, Input/output interfacing. 4. To impart knowledge in sensors and actuators. 5. To familiar with the real world application development using embedded system.
	<p>Course Outcome:</p> <p>CO1: Differentiate between general purpose and embedded systems</p> <p>CO2: Discuss the characteristics and quality attributes of embedded systems</p>

	CO3: Use different types of sensors for appropriately CO4: Design and develop embedded systems
Computer Oriented Statistical Techniques	
Course Code: USIT403	Course Objectives: <ol style="list-style-type: none"> 1. To learn the different methods of calculating the central tendencies. 2. To introduce the moments, skewness and kurtosis. 3. To learn scientific view to conduct the survey in proper way to collect the data about specific perspective. 4. To Learn variety of probability sampling methods for selecting a sample from a population. 5. To learn the sampling theory and testing of hypothesis and making inferences. 6. To introduce the students with understanding of the curve fitting, regression and correlation techniques.
	Course Outcome: Upon the successful completion of the course, students will be able to: CO 1: To calculate and apply measures of central tendencies and measures of dispersion -- grouped and ungrouped data cases. CO 2: To calculate the moments, skewness and kurtosis by various methods. CO 3: How to apply discrete and continuous probability distributions to various business problems. CO 4: Perform Test of Hypothesis as well as calculate confidence interval for a population parameter for single sample and two sample cases. Understand the concept of p-values CO 5: Apply simple linear regression and correlation model to real life examples.
Software Engineering	
Course Code: USIT404	Course Objective: <ol style="list-style-type: none"> 1. Develop the software projects or prototypes by understanding the requirements. 2. Meet the project deadlines along with the number of resources and type of tasks to be carried out. 3. Evaluate and analyze the SDLC and basic architecture SRS documents. 4. Help to understand the software design and coding techniques. 5. Understand the software testing principles. 6. Understand the concept project management. 7. Identify various concepts of Advanced UML techniques
	Course Outcome: After completing the course, the learner will be able to: CO1: Understand software engineering CO2: Apply software engineering principles CO3: Discuss various approaches to verification and validation of software including testing, measurements and estimation of software products CO4: Create software using different software development models
Computer Graphics and Animation	

<p>Course Code: USIT405</p>	<p>Course Objectives:</p> <ol style="list-style-type: none"> 1. To train the students to acquire skills in generating marketable computer graphics and animated pictures, especially in the area of advertisements. 2. To train the students to acquire skills and mastery in the use of different software producing graphics and animation. 3. The course introduces the basic concepts of computer graphics. 4. It provides the necessary theoretical background and demonstrates the application of computer science to graphics. 5. The course further allows students to develop programming skills in computer graphics through programming assignments.
	<p>After completion of the course students are supposed to be able to:</p> <p>CO 1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics</p> <p>CO 2. Compare various algorithms for scan conversion and filling of basic objects</p> <p>CO 3. Use of geometric transformations on graphics objects and their application in composite form.</p> <p>CO 4. Extract scene with different clipping methods and its transformation to graphics display device.</p> <p>CO 5. Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.</p> <p>CO 6. Render projected objects to naturalize the scene in 2D view and use of illumination models</p> <p>CO 7. Understand the core concepts and mathematical foundations of computer graphics</p> <p>CO 8. Know the fundamental computer graphics algorithms and data structures</p> <p>CO 9. Understand an overview of different modeling approaches and methods</p> <p>CO 10. Apply basic shading and texture mapping techniques</p> <p>CO 11. Understand light interaction with 3D scenes</p> <p>CO 12. Explain the applications, areas, and graphic pipeline, display and hardcopy technologies.</p> <p>CO 13. Apply and compare the algorithms for drawing 2D images also explain aliasing, antialiasing and half toning techniques.</p> <p>CO 14. Discuss OpenGL application programming Interface and apply it for 2D & 3D computer graphics.</p> <p>CO 15. Analyze and apply clipping algorithms and transformation on 2D images.</p> <p>CO 16. Solve the problems on viewing transformations and explain the projection and hidden surface removal algorithms.</p> <p>CO 17. Apply basic ray tracing algorithm, shading, shadows, curves and surfaces and also solve the problems of curves.</p>

