



Faculty of Computer Science

Master of Science (Information Technology) - (M.Sc.I.T.)

(2 years – Four Semester Full Time Course)

Semester: II Subject Code: MSIT201 Name: OBJECT ORIENTED CONCEPTS & C++ PROGRAMMING

Sr. No.	Subject Code	Name of the Subject	Teaching Scheme (Hours)				Evaluation Scheme								
			Th	Tu	Pr	Total	Theory					Practical (Marks)			Total
							Sessional Exam		University Exam		Total	Pr/Viva	T W	Total	
							Mark s	Hr s	Mark s	Hr s					
1	MSIT201	OBJECT ORIENTED CONCEPTS & C++ PROGRAMMING	4		2	6	30	1.5	70	2.5	100	40	10	50	150

Objectives:

- ✚ The C++ language most demanding language as a tool for all types of work. How this important language is to be mastered and how to use this knowledge in building efficient and flexible code is one of the prime requirements today.
- ✚ The course helps to the students to improve the object oriented programming skills.

Prerequisites:

- ✚ Knowledge of C programming
- ✚ Programming concepts including algorithm designing and logic.

Course Outline:

Sr. No.	Course Contents	Number of Hours
1	Introduction to Object Oriented Concepts Object Oriented Concepts, Object, Class, Keywords, Identifiers, Data types,	4



	Constants, Features of C++, Differentiate Object Oriented V/s Procedure Oriented	
2	Overview of C++ Language Operators in C++, Conditional structure and looping structure, Differentiate struct v/s class, Differentiate union v/s class, Application of pointer in object oriented concepts, Pointer to objects and pointer to members of class, The local classes, Assigning objects	5
3	Functions Utility in object oriented Approach Function Introduction, The inline function, Default arguments to the function, Object as a parameter, Call by reference and return by reference, Function Prototyping, Function overloading, Friend Function, utility of friend function with examples, Constant and volatile function, Static function, Private and public function, Function using pointer	5
4	Application of Constructors & Destructors in Object oriented Concepts Constructor, Application of Constructor & Rule to define the constructor, Types of Constructor, Explicit constructor, Parameterized constructor, Multiple Constructor(With Example), Dynamic Initialization, Constructor with dynamic allocation, Copy constructor.	4
5	Operator Overloading & User define function: Arithmetic operator overloading, Unary , Binary Operator Overloading, Assignment Operator Overloading, Subscript operator overloading, Operator overloading with Friend Function, The need for user defined conversion, Four different cases where user defined conversions are needed, Comparison of both the methods of conversion.	5
6	Templates Use of Templates, Define Function Templates, Function Templates with Generic & Non Generic Types, Define Class Templates, Specialization In templates, Define Class and Generic Data Types, Static Data Member in Templates. Export, typename Keyword.	5
7	Inheritance Application of Inheritance, Defining derived class using single base class, Define Different Types of Derivation using Access modifiers, The implementation of inheritance in the C++ object model,	3



	The Access Control, Declaration, The multiple-inheritance, Abstract classes, Composite objects	
8	Runtime Polymorphism: Difference Between Compile time and Run time polymorphism, Pointers to Objects, This pointer, Compatibility of Derived and base class pointers, The sub object concept, Virtual functions, Static invocation of virtual function, Default arguments to virtual functions, Virtual destructors, Pure virtual functions, RTTI.	4
9	Exception Handling Introduction, Exception Handling, Mechanism, Try, Catch and throw mechanism, Re throwing an exception, Terminate and Unexpected functions, Drawbacks of exception handling approach, The exception Class	3
10	IO Streams Stream, Difference of C and C++ IO Stream, The C++ Predefined streams, Formatting IO, IOS Members, Manipulators, Creating own manipulator	4
11	Using Files for IO Why IO is special, Different File Modes, File Handling, Create, Update, Delete, Files, Random Access using seek, IO Modes, Handling File Control Errors	4
12	Namespaces Introduction and need of name space, Defining namespaces, Extending the namespace	2
13	The Standard Template Library Introduction, Generic Programming Technique, Generic Software Designing technique, Components, Generic Algorithms, Iterators, Containers, Algorithms	2

List of Practical:

1.	Write a C++ program to find the sum of individual digits of a positive integer.
2.	A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.
3.	Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
4.	Write C++ programs that use both recursive and non-recursive functions



	<p>a. To find the factorial of a given integer.</p> <p>b. To find the GCD of two given integers.</p> <p>c. To find the nth Fibonacci number.</p>
5.	Write a C++ program that uses a recursive function for solving Towers of Hanoi problem.
6.	Write a C++ program that uses functions
7.	Write a C++ program to find both the largest and smallest number in a list of integers.
8.	Write a C++ program to sort a list of numbers in ascending order.
9.	Write a C++ program that uses function templates to solve problems-7&8.
10.	Write a C++ program to sort a list of names in ascending order.
11.	Write a C++ program to implement the matrix ADT using a class. The operations supported by this ADT are:
12.	Implement the matrix ADT presented in the problem-11 using overloaded operators (<<, >>, +, -, *) and templates.
13.	Implement the complex number ADT in C++ using a class. The complex ADT is used to represent complex numbers of the form $c=a+ib$, where a and b are real numbers. The operations supported by this ADT are:
14.	Write a C++ program that overloads the + operator and relational operators (suitable) to perform the following operations:
15.	Implement the complex number ADT in C++ using a class. The complex ADT is used to represent complex numbers of the form $c=a+ib$, where a and b are real numbers. The operations supported by this ADT are:



C. U. SHAH UNIVERSITY

	<p>c) Addition of Complex numbers. f) Division of complex numbers.</p> <p>Note: 1. overload << and >> operators in part a and part b.</p> <p>2. overload +, -, *, / operators in parts c, d, e and f.</p>
16.	Write a template based C++ program that determines if a particular value occurs in an array of values.
17.	Write a C++ program that uses functions to perform the following operations: a. Insert a sub-string into the given main string from a given position. b. Delete n characters from a given position in a given string.
18.	Write a C++ program that uses a function to reverse the given character string in place without any duplication of characters.
19.	Write a C++ program to make the frequency count of letters in a given text.
20.	Write a C++ program to count the lines, words and characters in a given text.
21.	Write a C++ program to determine if the given string is a palindrome or not.
22.	Write a C++ program to make frequency count of words in a given text.
23.	Write a C++ program that displays the position or index in the string S where the string t begins, or -1 if S doesn't contain t.
24.	2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C++ program to find the 2's complement of a binary number.
25.	Write a C++ program that counts the number of 1 bit in a given integer.
26.	Write a C++ program to generate Pascal's triangle.
27.	Write a C++ program to construct of pyramid of numbers.
28.	Write a C++ program to compute the Sine series.
29.	Write a C++ program that converts Roman numeral into an Arabic integer.
30.	Write a C++ program which converts a positive Arabic integer into its corresponding Roman Numeral.
31.	Write a C++ program to display the contents of a text file.
32.	Write a C++ program which copies one file to another.
33.	Write a C++ program to that counts the characters, lines and words in the text file.
34.	Write a C++ program to change a specific character in a file.
35.	Write a C++ program to reverse the first n characters in a file.
36.	Write a C++ program that uses a function to delete all duplicate characters in the given string.
37.	Write a C++ program that uses a function to convert a number to a character string.



38.	Write a C++ program that uses a recursive function to find the binary equivalent of a given non-negative integer n.
39.	Write a C++ program to generate prime numbers up to n using Sieve of Eratosthenes method.
40.	Write a C++ program a) To write an object to a file. b) To read an object from the file.
41.	Write C++ programs that illustrate how the following forms of inheritance are supported: a) Single inheritance b) Multiple inheritance c) Multi level inheritance d) Hierarchical inheritance
42.	Write a C++ program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class.
43.	Write a C++ program that illustrates how run-time polymorphism is achieved using virtual functions.
44.	Write a C++ program that illustrates the role of virtual base class in building class hierarchy.
45.	Write a C++ program that illustrates the role of abstract class in building class hierarchy.

Learning Outcomes:

- ✚ Students should be able to understand and appreciate the Object Oriented approach of Programming
- ✚ Students should be able to solve problems given to him/her using C++ with keeping balance between efficiency and flexibility

Books Recommended:

- ✚ “Object Oriented programming with C++”, E. Balagurusamy, TMH
- ✚ “Complete Reference C++“, Herbert Schildt McGraw Hill Publications
- ✚ “Computer Science- A Structured approach using C++”, Forouzan ,Gilburg, THOMSON Books
- ✚ “Object Oriented programming in C++”, Robert Lafore, Pearson Education
- ✚ “C++ Primer”, Stanley Lippmann, Pearson Education
- ✚ “The C++ Programming Language”, Bjarne Stroustrup, Pearson Education
- ✚ “Effective C++”, Scott Mayer Addison Wesley
- ✚ “OOP with C++”, S .Sahay, Oxford Higher Education.
- ✚ “C++ and OOP Paradigm”, D.Jana, 2nd Edition, PHI.



Faculty of Computer Science

Master of Science (Information Technology) - (M.Sc.I.T.)

(2 years – Four Semester Full Time Course)

Semester: II

Subject Code: MSIT202

Name: JAVA PROGRAMMING

Sr. No.	Subject Code	Name of the Subject	Teaching Scheme (Hours)				Evaluation Scheme								
			T h	T u	P r	Tota l	Theory					Practical (Marks)			Total
							Sessional Exam		University Exam		Tota l	Pr/ Viv a	T W	Tot al	
							Mark s	Hr s	Mark s	Hr s					
1	MSIT202	JAVA PROGRAMMING	4		4	8	30	1.5	70	2.5	100	80	20	100	200

Objectives:

- ✚ To develop proficiency in creating console based and GUI based applications using the Java Programming Language.
- ✚ To be able to understand the concepts of Object Oriented Programming Language and easily use Java.
- ✚ To get a good understanding of developing multi-threaded applications using the Java Programming Language.
- ✚ To be able to develop Applets for embedding in a web page.

Prerequisites:

- ✚ Knowledge of Algorithm and Flow chart to implement the programming logic.

Course Outline:

SNo.	Course Contents	Number of Hours
1	Introduction Introduction – what is java, importance of java, java implementation application of java, java buzzwords (simple, secure, portable, object-oriented, robust	6



	<p>multithreaded, architecture – natural, interpreted, high performance, distributed dynamic) object oriented programming</p> <p>three OOP principals (encapsulation, inheritance, polymorph) sample Program & compilation, block of code, lexical issues (White space, identifiers, literals, comments, separators, keyword), java class library</p>	
2	<p>Data type, operators, control structures</p> <p>variables, constants, declaration, literals, scope of variable, type casting</p> <p>arithmetic operators, relational operators, logical operators, assignment operators, increment –decrement operators, conditional operators, bit wise operators, interface of operators, dot operators</p> <p>if-else, statement, loops (while, do-while, for break, go to, continue return)</p> <p>switch statement, operator</p> <p>arrays –declaration, creation, initialization, length</p> <p>two-dimensional arrays</p> <p>string-string arrays, string methods, string buffer class</p>	4
3	<p>Introduction of classes, objects and methods</p> <p>class, object & method</p> <p>defining class, adding variables, adding methods, creating objects</p> <p>constructors THIS key word, garbage collection, finalize() method</p> <p>accessing class members, methods overloading static members, nesting of methods</p> <p>vectors and wrapper classes</p> <p>inheritance, subclasses, subclass constructor, multiple inheritance, hierarchical inheritance overriding methods</p> <p>final variables and methods, final classes, finalize methods, abstract methods and classes</p> <p>visibility control – public access, friendly access, protected access, private protected access, rules of thumb</p> <p>method overloading, object as parameters, argument passing, returning objects, recursion, access control, static, final</p> <p>Nested & inner classes</p> <p>String class , Command-Line arguments</p>	10
4	<p>Inheritance, Packages and Interfaces</p> <p>Inheritance, Member access, super class creating multilevel Hierarchy</p>	6



	<p>Method over loading & overriding, Abstract class, method</p> <p>Using final to prevent overriding & overloading, the object class</p> <p>Defining packages, understanding CLASSPATH ,Access protection</p> <p>importing packages, defining interfaces</p>	
5	<p>Exceptions, Nested enum types and Collection framework</p> <p>exception types, uncaught exceptions</p> <p>multiple catch clauses ,nested try statements</p> <p>throw, throws, finally, java's built-in exceptions</p> <p>creating your own exception subclasses</p> <p>Member Types, Top level nested classes and Inner classes, The local class and anonymous classes, The enum type, classes from java.util package</p> <p>Date, Time Zone, Calendar and the Gregorian Calendar classes</p> <p>Collection Framework, Collection interface, Set and List interfaces, Map interface</p>	8
6	<p>I/O files in java, Multithreaded programming</p> <p>Concept of streams, difference between characterstreams and bytestreams</p> <p>characterstreams(reader,writer,bufferedReader,inputstreamreader,fileReader,bufferWriter, outputStreamreader, filewriter, printwriter)</p> <p>Bytestream(inputstream, fileinputstream, filterinputstream, bufferedinputstream, datainputstream, outputStream, fileoutputstream, filteroutputstream, bufferedoutputstream, dataoutputstream, printstream)</p> <p>Other classes (random access file , stream tokenizer)</p> <p>creating threads, run()method, new thread, thread class, stopping &blocking threads, life cycle of thread- newborn, runnable, running, blocked, dead, waiting sleeping, suspended, blocked, using thread methods, thread exceptions, thread priority, synchronization, implementing the Runnable interface</p>	6
7	<p>Applet, Event Handling</p> <p>Introduction to applet, applet lifecycle ,applet class</p> <p>applet context class, passing parameters to applet</p> <p>use of java .awt graphics class and its various methods in an applet</p> <p>Event delegation model or event class hierarchy</p> <p>all classes and interfaces of event delegation model, programmers related to event handling covering all types of events</p>	4
8	<p>Graphical user interface, Java Swing</p>	4



Layout managers (flowlayout, borderlayout, cardlayout gridbaglayout, gridlayout) AWT controls (labels, buttons, canvases, checkboxes, checkboxgroup, choices, textfields, textareas, lists, scrollbars, panels, windows, frames, menus, menubars)	
Total Lecture	48

List of Practical:

SNo.	Course Contents
1	Write program for simple print “Wel come” in screen
	Write program for find Odd and Even number
	Program- Write a java program to calculate Factorial of given no through command line argument
2	Write a java program to calculate area of circle ,use command line arg to accept the value of radius
	Program that accepts two Double numbers as its command line args. Multiply these together and display the Product.
3	Program that defines a circle class with two constructors. The first from accepts a double value that represents the radius of circle. This constructor assumes that the circle is centered at the origin. The second form accepts the double value & the first two arguments define the co-ordinate of the center and the third arguments define the radius.
	program to sort the element of an array in ascending order using command line argument
4	Program that accepts two Double numbers as its command line args. Multiply these together and display the Product.
	Program that outputs a table of numbers each line in the table contains three entries the number its cube begin with 1 and end with 10 also use a FOR loop to generates.
	/* Display the following outputs 1 1 2 2 1 2 3 3 3 1 2 3 4 4 4 4 1 2 3 4
5	Write a java program to find power of given number use command line argument to accept base and power number
	Write a program for print the series like 1 + 1/2 + 1/3....
	Write a program for print the Fibonacci series
6	Program to create a STRINGBUFFER object and illustrate how to insert character as its beginning



	<p>Program to create an application which will read string from command line args and will return into alphabetical order.</p> <p>EX. string:- AMPICS O/P:- ACIMPS</p>
7	<p>Program that display cos,sin and tan to its command line args. however it is displayed an error message if no command line args has been provided</p> <p>Program that searches through its command line args if an argument is found that does not begin with an Upper case letter. Display an error message and terminate</p>
8	<p>Create multiple threads.</p> <p>Create package with sum of three class</p> <p>Program to print words,lines,caharacters in a file</p>
9	<p>Program to print information about a file.</p> <p>Program to print sum of integers using StringTokenizer.</p> <p>Write a java program to read a text and count the occurrences of word</p>
10	<p>Program for Applet Life Cycle with appropriate Message</p> <p>Design Indian Flag.</p> <p>program create an apptel which has two buttons red and green. create a event when red button is pressed the background of the applet will be red and also green respectively</p>
11	<p>Program to create a circle on the center of the applet and fill color with magenta</p> <p>Write a Applet program to show NAME and PASSWORD label and textbox resp. to enter text in it.</p>
12	<p>Write an applet that accepts multiple parameters that identifying a set of images select one of these images at random and display it</p> <p>Write an applet that accepts multiple parameters that identifying a set of images select one of these images at random and display it</p>
13	<p>Write a Java program that allows the user to draw lines, rectangles and Ovals.</p> <p>Write a java program that handles key events.</p>
14	<p>Write a java Applet to display nested layout</p> <p>Write a java applet program of scrolling list with choice & inform user to its select it</p>
15	<p>Program To Create a File Menu</p>
16	<p>Demonstrate the mouse event handlers.</p>
17	<p>Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the + - X % operations. Add a text field to display the result.</p>
18	<p>Write an applet that computes the payment of a loan based on the amount of the loan, the interest rate and the number of months. It takes one parameter from the browser: Monthly rate; if true, the interest</p>



	rate is per month; Other wise the interest rate is annual.
19	Program to multiply two matrices.
20	Write an application that create a frame with one button each time button is pressed a model dialog box appears that displays the current date.

Learning Outcomes:

- ✚ Ability to create appropriate classes using the Java Programming Language to solve a problem using Object Oriented Approach.
- ✚ Ability to write console based and GUI based applications in the Java Programming Language.
- ✚ Ability to develop to multi-threaded applications using the Java Programming Language
- ✚ Ability to create Applets using the Java Programming Language

Teaching & Learning Methodology:

- ✚ Using Whiteboard & Multimedia or OHP

Books Recommended:

- ✚ “Java: the Completed Reference , 7th Edition by Schildt, Herbert, TMH publication
- ✚ “The class of Java”, Pravin Jain, Pearson Education.
- ✚ “Programming with Java a Primer 3e”, Balagurusamy, McGraw Hill
- ✚ “The Java Programming Language”, Ken Arnold, James Gosling, David Holmes , Addison-Wesley Pearson Education (4th Edition – 2005).
- ✚ “Object-Oriented Programming with Java: Essentials & Applications”, Raj Kumar Buyya, S. Thamarai Selvi, & Xing Chen Chu, Tata McGraw Hill



Faculty of Computer Science

Master of Science (Information Technology) - (M.Sc.I.T.)

(2 years – Four Semester Full Time Course)

Semester: II

Subject Code: MSIT203

Name: SOFTWARE ENGINEERING

Sr. No.	Subject Code	Name of the Subject	Teaching Scheme (Hours)				Evaluation Scheme								
			T h	T u	P r	Tota l	Theory					Practical (Marks)			Total
							Sessional Exam		University Exam		Tota l	Pr/ Viv a	T W	Tot al	
							Mark s	Hr s	Mark s	Hr s					
1	MSIT203	SOFTWARE ENGINEERING	4			4	30	1.5	70	2.5	100				100

Objectives:

- To be able to understand the concepts of Designing Software and Getting Cost Estimations

Prerequisites:

- Knowledge of Basic System Analysis and Design

Course outline:

Sr. No.	Course Contents	Number of Hours
1	Introduction to Software Engineering, Process and Process Models Introduction to Software Engineering, Evolving Role of Software, Legacy Software. A Layered Technology, A Process Frame Work, The Process Pattern. Prescriptive Models, The Waterfall Model, The Serialized and Unified Process.	10
2	Requirements Engineering Problem Recognition, Requirement Engineering tasks, Processes, Requirements Specification, Use cases and Functional specification, Requirements validation, Requirements Analysis, Modeling – different types	10



3	Object Oriented Analysis and Design Object Oriented Analysis Concept, Domain Analysis, Generic Concept of Object Oriented Analysis Model, Object Oriented Analysis Process, Object Relationship Model, Object Behavior Model. Design of Object Oriented System, The System Design Process, Object Design and System Design Process	10
4	Testing Strategies and Tactics A Strategic Approach to Software Testing (Verification and Validation) Strategic Issues, Validation Testing (Criteria, Configuration Review, alpha and beta Testing), The art of Debugging (Debugging Process, Strategies, Correcting the Error) , Software Testing Fundamentals, Black Box and White Box Testing, Object Oriented Testing Methods.	10
5	Clean Room Software Engineering and Component Base Software Engineering The Clean Room Approach, Functional specification, Clean room specification Clean room design, Clean room testing, Engineering of component based systems The component based software engineering process Domain engineering, Component based development Classifying and Retrieving Components	8
Total hours		48

Learning Outcomes:

- ✚ He/She should be able to understand and appreciate the Web Technology.
- ✚ He/She should be aware of the working and architectural Web Site.
- ✚ He/She should be able to solve problems given to him/her using PHP efficiency.

Books Recommended:

- ✚ “Software Engineering – A practitioner’s Approach”, Roger Pressman. 6th Edition.
- ✚ “Object Oriented Analysis and Design” Gooch
- ✚ “Fundamentals of Software Engineering”, Rajib Mall.



Faculty of Computer Science

Master of Science (Information Technology) - (M.Sc.I.T.)

(2 years – Four Semester Full Time Course)

Semester: II

Subject Code: MSIT204

Name: OPERATING SYSTEM

Sr. No	Subject Code	Name of the Subject	Teaching Scheme (Hours)				Evaluation Scheme								
			T h	T u	P r	Tota l	Theory					Practical (Marks)			Total
							Sessional Exam		University Exam		Tota l	Pr/ Viv a	T W	Tot al	
							Mark s	Hr s	Mark s	Hr s					
1	MSIT204	OPERATING SYSTEM	4		2	6	30	1.5	70	2.5	100	40	10	50	150

Objectives:

- ✚ Help students become familiar with the fundamental concepts of operating system.
- ✚ Help students become competent in recognizing operating systems features and issues.
- ✚ Provide students with sufficient understanding of operating system design and how it impacts application systems design and performance.

Prerequisites:

- ✚ Basics of Computer System Architecture
- ✚ C / C++ Programming Skills

Course Outline:

SNo.	Course Contents	Number of Hours
1	Introduction Evolution of operating systems. Types of operating systems. Different Views of the operating systems, operating system concepts and structure.	4
2	Processes The Process concept, systems programmer’s view of processes. The operating system	6



	services for process management, Scheduling algorithms. Performance evaluation.	
3	Memory Management Memory Management without swapping of paging, swapping, virtual memory page replacement algorithms, modeling paging algorithms, design issued for paging systems, segmentation.	8
4	Inter-process Communication and Synchronization The need for interprocess synchronization, natural exclusion., semaphores, hardware sport for mutual exclusion, queuing implementation of semaphores, classical problems in concern programming critical region and conditional critical region, monitors, messages deadlocks.	8
5	File Systems: Field systems, directories, file systems implementation, security protection mechanism	6
6	Input/Output Principles of I/O Hardware: I/O devices, devices controllers direct memory access. Principles of I/O Software: Goals, interrupt handlers, device drivers, device independent I/O software, User space I/O software. Disks. Disk hardware, scheduling algorithms, Error handling, trac-at-a-time caching, RAM Disk.	10
7	Information Management: A Simple File System; General Model of a File System; Symbolic File System; Basic File System, Access Control Verification; Logical File System; Physical File System; Case study on DOS, Windows 98, Windows NT & Linux.	6
Total Lecture		48

List of Practical:

SNo.	Practical Description
1	Study of laboratory environment: Hardware specifications, software specifications Some UNIX commands: date, ls, who, cal, ps, wc, cat, uname, pwd, mkdir, rmdir, cd, cp, rm, mv, diff, chmod, grep, sed, head, tail, cut, paste, sort, find.
2	Simple Unix-C programs: Programs using system calls, library function calls to display and write strings on standard output device and files.
3	Programs using fork system calls.



4	Programs for error reporting using <code>errno</code> , <code>perror()</code> function.
5	Programs using pipes.
6	Shell Programming
7	Programs to simulate process scheduling like FCFS, Shortest Job First and Round Robin.
8	Programs to simulate page replacement algorithms like FIFO, Optimal and LRU.
9	Programs to simulate free space management.
10	Programs to simulate virtual memory.
11	Programs to simulate deadlock detection.

Learning Outcomes:

- ✚ Upon successful completion of this course, the student shall be able to:
- ✚ • Exhibit familiarity with the fundamental concepts of operating systems.
- ✚ • Exhibit competence in recognizing operating systems features and issues.
- ✚ • Apply a mature understanding of operating system design and how it impacts application systems design and performance.

Teaching & Learning Methodology:

- ✚ Using Whiteboard & Multimedia or OHP

Books Recommended:

- ✚ “Operating Systems”, Madnick E., Donovan J., Tata McGraw Hill.
- ✚ “Operating System Concepts”, Silbershatz and Galvin, Addison Wesley.
- ✚ “Operating System Concepts”, Peterson, J.L. Abraham Silberschatz, Addison Wesley publishing
- ✚ “Unix Systems Programming : Communication, Concurrency and Threads”, Kay Robbins, 2-Edition, Pearson Education
- ✚ “Unix concepts and applications”, Sumitabha Das, TMH Publications.
- ✚ “Unix programming”, Stevens, Pearson Education.



Faculty of Computer Science

Master of Science (Information Technology) - (M.Sc.I.T.)

(2 years – Four Semester Full Time Course)

Semester: II

Subject Code: MSIT205

Name: WEB TECHNOLOGY

Sr. No	Subject Code	Name of the Subject	Teaching Scheme (Hours)				Evaluation Scheme								
			Th	Tu	Pr	Total	Theory					Practical (Marks)			Total
							Sessional Exam		University Exam		Total	Pr/Viva	TW	Total	
							Mark s	Hr s	Mark s	Hr s					
1	MSIT205	WEB TECHNOLOGY	4		2	6	30	1.5	70	2.5	100	40	10	50	150

Objectives:

- ✚ The internet has drastically changed the way we communicate. As web technology dissolves the world’s borders, a new “global community” has emerged.
- ✚ The course will focus on methods of using interconnected networks to effectively distribute text and information.
- ✚ The course will focus on overall site design strategies, explore web usability/interface problems, and outline effective solutions.
- ✚ Students will learn and implement HTML to construct a website with consideration to course topics.
- ✚ We seek an advanced mastery of web-development techniques that use databases to create content—HTML form objects, database connections, and server-side programming. We will use open-source MySQL as our database, structured query language (SQL), and PHP for programming

Prerequisites:

- ✚ The course is for advanced students with career or program-related needs for Web applications training.
- ✚ Students should be familiar with Windows operating systems and with technology for static web pages.



Course Outline:

Sr. No.	Course Contents	Number of Hours
1	Introduction to Javascript Introduction to Javascript, Features, Writing Methods in HTML, Data Types, Variable Creations, Array, Operators, Conditional Checking, Looping Structures, UDF, Dialog Boxes, Built-In Objects (String, Math, Date), Cookies.	8
2	Introduction to PHP Introduction to PHP, Features, Installation of IIS, Variable Declaration (Static, Global), Operators and Expressions, Decision Making, Looping Structures, Arrays, UDF (argument function, default function, return function), Variable Functions (Gettype, settype, intval, print_r, strval, floatval, isset, unset) String Functions, Math Functions, Date Functions, Array Functions, Miscellaneous Functions, File Handling Functions.	8
3	Component of PHP PHP Regular Expression, Cookies, Session, GD Library	5
4	XML with PHP Introduction to XML, XML Document Structure, Creating XML File, Root and Child Node concept, XML Elements and Attributes, The SimpleXML Extension.	5
5	Database Programming Introduction to MySQL (Using PHP MyAdmin), PHP MySQL Connectivity, Basic Connection Functions, Handling Server Errors, Creating Database, Tables, Insert Data into Tables, Retrieving data from MySQL, Retrieving Fields	6
6	Object Oriented Programming Introduction to OOP, Classes, Objects, Inheritance, Constructor, Serialized Object, Overloading, Encapsulation	8
7	AJAX with PHP Introduction to AJAX, Server Side Scripting Technology, Request and Response Concept, Creating Web Page with AJAX, AJAX with Database	8
Total hours		48

List of Practical:



Sr. No.	Course Contents
1	Design a web page using different text formatting tags
2	Design a web page with links to different pages and allow navigation between pages
3	Design a web page with Imagemaps
4	Design a web page with different tables. Design a webpage using table so that the content appears well placed
5	Design a webpage using frames
6	Design a web page with a form that uses all types of controls
7	Design a website using style sheets so that the pages have uniform style
8	Using Java Script design a web page that prints factorial / Fibonacci series / any given series
9	Design a form with a test box and a command button. Using Java Script write a program whether the number entered in the text box is a prime number or not.
10	Design a form and validate all the controls placed on the form using Java Script
11	Design a DTD, corresponding XML document and display it in browser using CSS
12	Design an XML document and display it in browser using XSL
13	Design XML Schema and corresponding XML document
14	Design a php page to process a form
15	Design a php page for authenticating a user
16	Design a complete dynamic website with all validations

Learning Outcomes:

Upon completion of this course, the student will be able to:

- ✚ write server-side scripts in the PHP language that process data from online forms and access MySQL databases to create dynamic Web pages.
- ✚ design and create 3-tier Web applications using PHP and MySQL.

Books Recommended:

- ✚ “Html, Dhtml, Javascript, Perl Cgi”, Ivan Byros, Bpb Publication
- ✚ “PHP and MySQL Web Development—Fourth Edition”, Luke Welling and Laura Thomson. Addison-Wesley.
- ✚ “Programming with Java”, Bhave, Pearson Education
- ✚ “PHP for the Web: Visual QuickStart Guide”, Ullman, Pearson Education
- ✚ “Java for programmers”, Deitel, Pearson Education.