NEW SCHEME FOR MASTER OF COMPUTER APPLICATOINS (MCA) COURSE

SEMESTER - I of MCA, Applicable from August 2020 onwards

SR. NO.	SUBJECT NO.	NAME OF THE SUBJECT	TEACHING SCHEME			EXAMINATION SCHEME							
			THEORY HR	TUTO HR.	PRACT. HR.	SESSIONAL M.	HR.	THEORY M.	HR	PRACT. M	H	T.W. MARKS	TOTAL
1	MCA 111	Bridge Course	4	-		-	-	-	-	-	-	-	P/F
2	MCA 112	Mathematical Foundations	3	1	-	25	2	50	3	-	-	25	100
3.	MCA 113	Object Oriented Concepts & Programming	4	-	3	25	2	50	3	50	3	25	150
4	MCA 114	Data Structures	4	-	3	25	2	50	3	50	3	25	150
5	MCA 115	Relational Database Management Systems	3	1	3	25	2	50	3	50	3	25	150
6	MCA 116	Elective - 1 (Any one subject from track 1/2/3)	4	-	3	25	2	50	3	50	3	25	150
	TOTAL		22	2	12	125	-	250	-	200	-	125	700

SUBJECTS FOR ELECTIVE - 1

(i) Introduction to Python Programming (Track-1)
(ii) Introduction to Linux Programming (Track-2)
(iii) Web Application Development (Track-3)

Gujarat University
Syllabus
MCA – I

Course Name: Mathematical Foundations 112

Course Code: MCA112

Objectives:

Mathematics is basic need in engineering, sciences, and computer applications. A few important applications are in logic development, data sciences, artificial intelligence & machine learning, and network security. It is very important to develop the student's geometric insight into the concepts of Calculus, Vectors, data representation and Linear Algebra and applying these concepts to real life problems and machine learning problems.

Aim of the course is to
· To introduce the concepts of calculus, vectors, data representation and linear algebra
· To apply these concepts to real life problems and machine learning problems

Prerequisites:
Basic knowledge of mathematical fundamentals

Contents

1. **Introduction to Set Theory**
   Basic Concepts, notations, inclusion and equality, power set, operations of union, intersection and complement, Venn diagrams, set identities (associative, distributive, etc), ordered pairs and n-tuples, Cartesian product

2. **Fundamentals of Single Variable Calculus**
   Functions of single variable, definition and their graphs, special functions like polynomials, trigonometric , exponential , hyperbolic , limit, continuity, definition of derivative and its graphical meaning , rules of differentiation, chain rule, higher order derivatives, definition of integration and its geometric interpretation ,indefinite and definite integral and their evaluation, Optimization of functions : Local Maxima and minima of functions, saddle point, , necessary and sufficient conditions, global maxima, convex functions, Taylor Series

3. **Graph and Tree**
   Basic concepts of Graph theory, paths, reachability and connectedness, matrix representation of graphs, trees.

4. **Fundamentals of Vectors**
   Definition of vector, scalars, addition and subtraction of vectors, scalar multiplication, inner product(dot product) of vectors, norms, direction, orthogonal vectors, projection of vectors, cosine similarity, normal and orthonormal vectors, Gram-Schmidt procedure, orthogonal decomposition

5. **Introduction to Matrices**
   Definition, addition of two matrices, transpose, scalar multiplication, matrix multiplication, properties of matrix addition and multiplication, square matrix , null and identity matrix , invertible matrix and inverse, hadamard product and its properties, determinant of a square matrix and its properties, rank , trace, popular type of matrices symmetric, diagonal, orthogonal, orthonormal

6. **Introduction to Numerical Methods**
   Introduction, Characteristic of numerical methods, Types and sources of errors in data, Quantification of errors, nature of iterative methods to find a solution, numerical methods of finding roots of an equation f(x) = 0 : Bisection method, false position method, Secant method, Newton Raphson Method , Gradient Descent method

7. **Linear Equations**
   Systems of Linear Equations, Cramer's Rule, Elementary row operations, row reduced and Echelon forms, Homogeneous Systems, Matrix inversion method

Reference Books:-

1. Advanced engineering Mathematics, 10'th Edition, ISV, Erwin Kreyszig, John Wiley and sons, INC.
2. Discrete mathematical structures with applications to computer science / J.P. Trembly, R. Manohar
3. Linear Algebra an its applications, 3'rd edition, David C. Lay, Pearson
4. Advanced Mathematics for engineers, Dr. Chandrika Prasad, Pothishala Private Ltd.

Accomplishments of the students after completing the course

1. Developing problem solving skills using mathematics techniques such as geometry, calculus and vectors.
2. Interpret, analyze and represent real life data into mathematical models.
3. Knowledge of Usage of mathematical models in various computer science domains such as artificial intelligence, network security, tend analysis and many more.

***** ***** *****

Course Name: Object Oriented Concepts and Programming

Course Code: MCA113

Objectives:

The programming for small devices like mobile phones, networking devices like routers, coding for graphics and multimedia, requires efficient coding as well as object oriented programming. The C++ language fits perfectly as a tool for this type 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 presented here is targeting to enable the student to master such skills. Aim of the course is to enable students to

1. Differente between procedural and object oriented programming.
2. Learn C++ as a language and various features of it.
3. Learn Object Oriented principles and their application using C++.

Prerequisites:

1. Knowledge of C language
2. Programming concepts including algorithm building and logic

Contents :

1. Introduction to C++, Overview of Core C++ Language, Classes and Objects [10%]

Identifiers and constants (Literals), Keywords, Data Types, The Operators, New Casting Operators, typeid and throw, The Conditional structures and Looping Constructs, , The Difference between struct and class in C++,The difference between Union and Class, Static Data members of a class, Pointer to objects and pointer to members of class, The local classes, Assigning Objects

2. Functions [10%]

Introduction, The inline function, Default Arguments to the function,Functions with object as parameters,Call by reference and return by reference, Prototyping and Overloading, Friend functions, Const and Volatile functions, Static functions, Private and Public functions, Function Pointers, Adding C functions to the C++ program

3. Constructors and Destructors [10%]

Introduction to constructors, The explicit constructors, Parameterized constructors, Having multiple constructors, Constructors with default arguments, Dynamic Initialization, Constructor with dynamic allocation, copy constructors, The member initialization list, destructors

4. Operator Overloading and User Defined Conversions [10%]

Introduction, Unary Operators, Binary Operators, Using Friends as operator functions, Overloading other Operators, The need for user defined conversion, Four different cases where user defined conversions are needed, Comparison of both the methods of conversion

5. Templates [5%]

Function Templates, Non Generic (Non Type) Parameters in Template functions, Template function and specialization, Overloading a template function, Using Default Arguments, Class Templates, Classes with multiple generic data types, Static data members, Primary and Partial Specialization, The Export Keyword, The other use of typename

6. Inheritance [15%]

The need, Defining derived class using single base class, Derivation using public, private and protected access modifiers, The implementation of inheritance in the C++ object model, The Access Control, The Access Declaration, The multiple-inheritance, Abstract classes, Composite objects (container objects)

7. Runtime polymorphism by virtual functions [10%]

Compile Time and Runtime Polymorphism, Pointers to Objects, This pointer, Compatibility of Derived and base class pointers, The subobject concept, Virtual functions, Static invocation of virtual function, Default arguments to virtual functions, Virtual destructors, Pure virtual functions

8. IO Streams [5%]

Need for streams, Advantages of using C++ I/O over C IO, The C++ Predefined streams, Formatting IO, Formatting using ios members, Manipulators, Creating our own manipulator

9. Using Files for IO [10%]

Why IO is special, Text and binary streams, Opening and closing files, Dealing with text files Dealing with binary files, Providing Random Access using seek, IO Modes, Handling Errors

10. Namespaces [5%]

Introduction and need, Use the using syntax, Defining namespaces, Extending the namespace, Unnamed namespaces, Nested Namespaces, Namespace aliases, The std namespace, The Koenig lookup, Overhead with namespaces

11. The Standard Template Library [5%]

The STL (Standard Template Library) Introduction, Generic Programming , Generic Software Components, Generic Algorithms, Iterators, Containers, Algorithms

Reference Book:

1. Programming with ANSI C++ by Bhushan Trivedi, Oxford University Press
2. Object-Oriented Programming with C++ by E Balagurusamy McGraw Hill Publications
3. C++ FAQs by Pearson Education
4. C++ Primer by Stanley Lippmann Pearson Education
5. The C++ Programming Language by Bjarne Stroustrup, Pearson Education
6. Effective C++ by Scott Mayer Addison Wesley
7. Complete Reference C++ by Herbert Schildt McGraw Hill Publications

Additional References :

ePgPathshala - Subject: Information Technology – P-01. Object Oriented Concepts and Programming using C++ http://epgp.inflibnet.ac.in/Home/ViewSubject?catid=305

Accomplishments of the student after completing the course :

1. He/She should be able to understand and appreciate the Object Oriented approach of programming
2. He/She should be aware of the working and architectural model of C++.
3. He/She should be able to solve problems given to him/her using C++ with keeping balance between efficiency and flexibility

***** ***** *****

Course Name: Data Structures

Course Code: MCA114

Objectives:

Students will be introduced to

- To develop proficiency in the specification, representation, and implementation of Data Types and Data Structures
- To be able to carry out the Analysis of various Algorithms for mainly Time and Space Complexity
- To get a good understanding of applications of Data Structures
- To develop a base for advanced computer science study

Prerequisites:

Any programming language like C, C++

Contents:

1. **Introduction**
   Data types, ADT, data structure: Definition & classification, Analysis of algorithms (recursive and non-recursive) with emphasis on best case, average case and worst case

2. **Structures, pointers & Dynamic Memory Allocation**
   Structures, Nesting of structures, Arrays of structures, Structures and pointers, Structures and functions, Union and its usage, Pointer - pointer arithmetic, Array and Pointers, Pointers and strings, Pointer to Pointer, Pointers and functions, Introduction to dynamic memory allocation, Allocating memory dynamically, resizing and releasing dynamically allocated memory

3. **Linear Data structures with applications**
   Array data structure: storage, mapping, applications (sparse matrix, polynomial representation, strings), List: Introduction, implementation using array & linked list (singly, doubly, circular, multi-list), Applications: Polynomial, representation, Sparse matrix, Stack: Introduction, implementation using array & linked list, Applications: Function call, Recursion, balancing of parenthesis, Polish Notation: infix to postfix conversion and evaluation of postfix expression, Queue: Introduction (queue, circular queue, deque, priority queue), implementation using array & linked list, Applications: Job Scheduling

4. **Non-Linear data structures**
   Tree: Introduction and representation, Forest, Tree traversal, Binary Tree (representation using array and links): Binary tree, traversal (recursive & non-recursive implementation), Expression tree, Graph: Introduction, representations, Traversal (BFS, DFS), Applications: Shortest path (Single source-all destinations), Minimal spanning tree (Prim's algorithm, Kruskal's algorithm)

5. **Searching and Sorting**
   Linear Search, Binary Search, Transpose sequential search, Binary search tree, Heap tree (application in priority queue and sorting), AVL tree, Splay tree, M-way search tree, B tree (insertion), B+ tree (Definition and introduction), B* tree (Definition and

introduction), Tries, Application of B tree and B+ tree in File Structures, Hash Tables: Introduction, hash functions and hash keys, Collisions, Resolving collisions, Rehashing, Sorting with algorithm analysis (best case, worst case, average): Bubble, Selection, Insertion, Shell, Merge, Quick, Heap, Radix

Notes:
Term work is to be carried out as per the above syllabus
Data Structures to be implemented in any programming language

References:

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education, 2nd edition (2003)
2. G. A.V. PAI, "Data structures and algorithms, concepts, Techniques and Applications", TMH, 1st Edition (2008)
3. Horowitz, Sahni, Anderson-Freed, “Fundamentals of Data Structures in C”, University Press (2nd edition-2007)
4. Jean-Paul Tremblay, Paul G. Sorenson, "An Introduction to Data Structures with Applications", Tata McGraw-Hill, 2nd Edition, (2007)
5. Cormen, Leiserson, Rivest, Stein, "Introduction to Algorithm", PHI (2003), 2nd Edition
6. Gilberg & Forouzan, "Data Structures: A Pseudo-code Approach with C", Thomson Learning.
7. Parag Dave & Himanshu Dave, "Design and Analysis of Algorithms", Pearson Education (2008)
8. Tanenbaum, "Data Structures Using C & C++", PHI.
9. Michel Goodrich, Roberto Tamassia, “Algorithm design-foundation, analysis & internet examples”, Wiley
10. A V Aho, J E Hopcroft, J D Ullman, "Data Structures & Algorithms", Addison-Wesley Publishing (1983).
11. Michael Berman, "Data Structures Via C++: Objects by Evolution", Oxford Univ. Press (2004)
12. D E Knuth, "Sorting & Searching - The Art of Computer Programming", Vol. 3, Addison- Wesley Publishing (1973)

Accomplishments of the student after completing the course:

After completion of this course, students will be able to:

- Ability to decide the appropriate data type and data structure for a given problem
- Ability to select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.
- The algorithms as referred above would include various operations on Queues, Stacks, Linked Lists, Trees, Graphs, Sorting, Searching, Hash tables
- Ability to compare algorithms with respect to time and space complexity

***** ***** *****

Course Name: Database Management Systems

Course Code : MCA 115

Objectives:

This course is intended to give students a solid background in database management systems and introduce NoSQL systems

Prerequisites:

Basics of Database Management Systems

Contents:

1. **Data Modeling/Conceptual Design & Relational Data Model**

   Data models: Introduction, Three level architecture, Overall architecture of DBMS, Various components of a DBMS, Enhanced ER diagrams, Database modeling using entity and relationships, Enhanced entity- relationship diagrams

2. **Relational Database Design**

   Relational structure - tables (relations), rows (tuples), domains, columns (attributes), keys: super key, candidate keys, primary key, entity integrity constraints, referential integrity constraints, Database design process, Anomalies in a database, Functional Dependencies(Lossless decomposition, Dependency preservance, Closure set of FD, Canonical cover, Lossless Joins), Finding Candidate keys using Armstrong rules, Stages of Normalisation: 1NF ,2NF ,3NF, BCNF (with general definition also) and Multi-valued Dependency : 4NF & 5NF(Project Join NF)

   Translation of E-R schemes (logical design) to relational schemes (physical design): A case study.

3. **Relational Algebra**

   Basic operators (Select, project, union, set, difference, cartesian product and rename) Additional operators (Set interaction, Natural Join, Division and Assignment operator), Insert, Update, Delete operators

4. **Transaction Management and Concurrency control**

   Transaction management: ACID properties, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks),Time stamping methods, optimistic methods, database recovery management.

5. **Database backup and Recovery**

   Need of Database backup, Database backup techniques, Types of Database failures, Types of Database recovery (Forward recovery, backward recovery and Media recovery),

6. **Security:**

   Introduction, Discretionary access control, Mandatory Access Control, Data Encryption

7. **Introduction to NoSQL**

What is NoSQL, Types of NoSQL Databases, How NoSQL Databases work, Distributed Databases, Introduction to Mongodb. SQL Vs NoSQL , Document Database , Use of JSON with document database, JSON in relational databases

8. **Introduction to other Databases**

Parallel Databases : Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems-Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism
Distributed Database Concepts : Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing
Multidimensional Databases and their uses in data analytics.
Temporal Databases : Introduction to Temporality, Temporal relationships, temporal hierarchies.
Spatial Databases: Spatial data types, spatial relationships, Topological Relationships, Spatial Data Structures and methods of storage.

PRACTICAL Only

9. **Query languages & Procedural SQL Practical only**

SQL Concepts : Basics of SQL, DDL,DML,DCL, structure – creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check, IN operator, Functions - aggregate functions, Built-in functions – numeric, date, string functions, set operations, sub-queries, correlated sub-queries, Use of group by, having, order by, join and its types, Exist, Any, All , view and its types. transaction control commands – Commit, Rollback, Savepoint.
PL/SQL Concepts : Cursors, Stored Procedures, Stored Function, Database Triggers

Main Reference Book(s):

1. Database System Concepts, S. Sudarshan , H.F. Korth, A. Silbershatz, McGraw Hill.
2. SQL - MySQL – MySQL 5 for Professionals , by Ivan Bayross
3. https://www.mongodb.com/nosql-explained
4. https://www.mongodb.com/document-databases

Suggested Additional Reading:

1. Database Management Systems, Ramakrishnan, Gehrke, McGraw Hill.
2. Database Systems: Design, Implementation and Management, Peter Rob, Carlos Coronel, Thomson Course technology.
3. Database Design . Application Development and Administration (Michael V. Mannino) - Third Edition McGraw Hill
4. Database Systems: Concepts, Design and Applications, Pearson Education, S.K.Singh.

Accomplishments of the student after completing the course:

Effective user or a DBMS Professional. A student would be able to effectively squeeze the "real world" data into the required data models of the database system and would be able to retrieve the data afterwards.

***** ***** ****

Course Name: Introduction to Programming with Python

Course Code: MCA 116 (T11)

Objectives:

This course will

- Introduce the core concepts of Python Programming
- Introduce the basic insight of programming using Python libraries and how to use functionality of various Python libraries for various tasks
- Give hands on with major focus on practical implementation of these concepts

Prerequisites:

Fundamentals of Computers

Contents:

1. **Introduction**
   Introduction to Python as a programming language, Introduction to various python programming editors (IDE), Python Virtual Machine, Memory Management in Python, Garbage Collector, Writing and Running Python programs, Built-in Data types, values and variables in Python, Evaluating Expressions and performing operations, Operators in Python, Conditional Execution in Python, Using iterations within Python programs, Input and Output in Python

2. **Functions in python**
   Introduction to Functions, Built-in Functions, Standard Functions: Mathematical Functions, time Functions, Random Numbers, System-specific Functions, The *eval* and *exec* Functions, Writing Functions, Function Basics Parameter Passing Documenting Functions, User defined Functions Vs. Standard Functions, Global Variables, Default Parameters, Introduction to Recursion, Functions as Data, Anonymous Functions, Function Decorators, Generators

3. **Arrays using array module and numpy() in python**
   Introduction to Array, Array using array module, Introduction to *numpy* module, Creating and importing Array, Types of Array, Indexing and slicing on Array, Mathematical operations on Array, Comparing Arrays, Aliasing Arrays, Attributes of an Array, Array methods, Basic operations on Multi-dimensional Array, Matrices in numpy, Random Numbers

4. **Lists, Tuples, Dictionaries & Sets in Python**
   Lists:
   Introduction to Lists, Use of List, Building Lists, List Membership, List Assignment and Equivalence, List Bounds, Slicing, List Element Removal, Lists and Functions, List Methods, Prime Generation with a List, Command-line Arguments, List Comprehensions, Multidimensional Lists, Lists Vs. Generators
   Tuples, Dictionaries, Sets:
   Tuples, Basic operations on Tuple, Nested Tuple, Dictionaries, Dictionary Methods, Counting with Dictionaries, Grouping with Dictionaries, Keyword Arguments, Sets, Set Quantification with all and any, Enumerating the Elements of a Data Structure

4. **Strings in Python**

Creating Strings, Indexing and Slicing in Strings, Concatenation of Strings, membership in Strings, Comparing Strings, Replacing a String, Splitting and Joining Strings, Changing case in Strings, String Testing methods, Formatting Strings, Sorting Strings, Substring, Inserting Substring into String

5. **Matrix in Python**
   Importing Matrix, basic operations on Matrix: finding maximum and minimum elements, Sum and average of elements, Products of elements, Sorting the Matrix, Transpose of a Matrix, Matrix operations, Diagonal elements of a Matrix, Random numbers

6. **Introduction to Object Oriented programming with Python**
   Introduction of OOPS, Classes and Objects: Creating Class, Constructor, Types of variable, types of Methods, Inner Classes, Encapsulation, Abstraction, Inheritance, Types of Inheritance, Overriding Super Class Constructors and Methods, super(), Method Resolution Order, Polymorphism, Duck typing philosophy of Python, Operator Overloading, Method Overloading, Method Overriding, Abstract Method, Abstract Class, Interface in Python, Abstract Classes vs. Interface

7. **Exceptions**
   Errors in Python, Exception and Exception handling, Types of Exceptions, *Except* block, *assert* statement, User-Defined Exceptions, Logging the Exceptions

8. **Files in Python**
   Files, Types of Files, Opening and Closing Files, Working with Text Files, Working with Binary *Files*, with statement, Pickle in Python, seek() and tell() methods, Random access of Binary Files, Zipping and unzipping Files

9. **Overview of Objects and Modules used with Python**
   Objects:
   Introduction to Objects, Introduction to Turtle Graphics Objects, Graphics with tkinter Objects, Date and time Objects, *import* statement, Introduction to *Pandas* and *Matplotlib* modules

References:

1. Rao N.R., “Core Python Programming”, Dreamtech Publication India
2. Sarker M.O.F., “Python Network Programming Cookbook”, Packt Publication
3. Halterman R.,”Fundamentals of Python Programming”, Southern Adventist University
4. Guttag J.V., “Introduction to Computation and Programming Using Python”, Prentice Hall India
5. Willi Richert, “Building Machine Learning Systems with Python”, Packt publication
6. Chun W., “Core Python Programming”, Prentice Hall India

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to gain awareness about various Python programming concepts, objects and libraries and able to solve challenging problems using Python programming language

***** ***** *****

Course Name: Linux Programming

Course Code: MCA 116 (T21)

Objectives:

- To get a good understanding of Linux internals.
- To develop proficiency in creating applications on Linux platform.
- To create a shell scripts for task automation

Prerequisites:

Knowledge of Operating Systems, C programming language

Contents:

1. **Linux Environment**
   A brief history of Linux, architecture of Linux, features of Linux, introduction to vi editor.
   Linux commands- PATH, man, echo, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, wc, lp, od, tar, gzip, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, unlink, du, df, mount, umount, find, unmask, ulimit, ps, w, finger, arp, ftp, telnet, rlogin.
   Text Processing utilities and backup utilities , tail, head , sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm, cmp, diff, tr, awk, cpio

2. **Introduction to Shell**
   Linux Session, Standard Streams, Redirection, Pipes, Tee Command, Command Execution, Command-Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment
   Customization Filters: Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count Characters, Words or Lines, Comparing Files,

3. **Shell Scripting**
   Uses of shell scripting, input and output to shell scripts, execution of shell scripts, positional parameters, loops and control structures, arrays and functions, creating automated tasks

4. **Process and Memory Management**
   Linux processes and signals, POSIX threads : creation, synchronization, attributes, canceling, Semaphores, shared memory and Message queues, Inter-process communication, Creating and using shared memory, Linux tools to debugging and make

5. **Socket Programming**

Socket, socket connections - socket attributes, socket addresses, socket, connect, bind, listen, accept, socket communications

Reference Books:

1) Neil Matthew, Richard Stones, "Beginning Linux Programming", Wrox Publication (Wiley India), 4th edition.
2) Venkateshmurthy M.G., Introduction to Unix and Shell Programming, Pearson India
3) Arnold Robbins, “Linux Programming by example – The Fundamentals”, Pearson Education.
4) Richard Stevens, “Advanced Unix Programming”, Pearson Education.
5) N.B. Venkateshwarlu, “Linux Programming Tools”, B. S. Publication – Hyderabad.
6) Eric S. Raymond, “The Art of Unix Programing”, Pearson Education.

Accomplishments of the student after completing the Course:

- Ability to develop console applications on Linux
- Ability to tweak Linux for optimum performance
- Ability to write scripts for recurring tasks

***** ***** *****

Course Name: Web Application Development

Course Code : MCA 116 (T31)

Objectives: Students will learn the platform neutral fundamentals of secure, dynamic web application development. Students will also learn how to implement a web application using one specific set of open sources server-side tools: PHP and MySQL.

Contents:

1. PHP Crash Course
Accessing PHP, Embedding PHP in HTML, Adding dynamic Content, Accessing Form Variables, Understanding Identifiers, Examining Variable Types, Declaring and Using Constants, Understanding Variable Scope, Using Operators, Using Variable Functions, Making Decision with Conditionals, Repeating Actions through Iteration, Using Declare

2. Storing and Retrieving Data
Saving Data for Later, Processing Files, Opening a File, Writing to a File, Closing a File, Reading from File, Using other Useful File Functions, Locking Files

3. Using Arrays
What is an array?, Numerically Indexed Arrays, Arrays with Different Indices, Array Operators, Multidimensional Arrays, Sorting Arrays, Sorting Multidimensional Arrays, Reordering Arrays, Loading Arrays from Files, Performing other array Manipulations

4. String Manipulation and Regular Expressions
Format Strings, Joining and Splitting Strings with String Functions, Compare Strings, Matching and Replacing Substrings with String Functions, Regular Expressions

5. Reusing Code and Writing Functions
The advantages of Reusing Code, Using require( ) and include ( ), Using Functions in PHP, Defining your own functions, Passing by Reference and Passing By Value, Implementing Recursions

6. Object-Oriented PHP
Understanding Object-Oriented Concepts, Creating Classes, Attributes and Operations in PHP, Implementing Inheritance in PHP, Understanding Advanced Object-Oriented Functionality in PHP

7. Error and Exception Handling
Exception Handling Concepts, The Exception Class, User-Defined Exceptions, Exceptions and PHP's Other Error Handling Mechanism

8. Designing Your Web Database
Relation Database Concepts, Designing Your Web Database, Web Database Architecture

9. Creating Your Web Database
Using the MySQL Monitor, Logging in to MySQL, Creating Databases and Users, Setting Up Users and Privileges, Introduction to MySQL's Privilege System, Setting Up a user for the Web, Creating Database Tables, Understanding MySQL Identifiers, Choosing Column DataTypes

10. Working with Your MySQL Database
What is SQL, Inserting data into the Database, Retrieving Data from the Database, Updating Records in the Database, Altering Tables After Creation, Deleting Records from the Database, Dropping Tables, Dropping a Whole Database

11. Accessing Your MySQL Database from the Web with PHP
How Web Database Architecture Work?, Querying a Database from the Web, Putting new information in the Database, Using Prepared Statements, Using Other PHP-Database Interfaces

12. Advanced MySQL Administration
Understanding the Privilege System in Detail, Making your MySQL Database Secure, Getting more Information About Databases, Optimizing Your Database, Backing Up your MySQL Database, Restoring Your MySQL Database, Implementing Replication

13. Advanced MySQL Programming
The LOAD DATA INFILE Statement, Transactions, Foreign Keys, Stored Procedures

14. Interacting with the File System and the Server
Uploading Files, Using Directory Functions, Interacting with the File System, Using Program Execution Functions,

15. Using Network and Protocol Functions
Sending and Reading Email, Using Data from Other Websites, Using Network Lookup Functions, Backing Up or Mirroring a File

16. Managing the Date and Time
Getting the Date and Time from PHP, Converting Between PHP and MySQL Date Formats, Calculating Dates in PHP, Calculating Dates in MySQL, Using Microseconds, Using the Calendar Functions

17. Using Session Control in PHP
What is Session Control?, Understanding Basic Session Functionality, Implementing Simple Sessions, Creating a simple session, Configuring Session control, Implementing Authentication with Session Control

Main Reference Book(s):

1. Luke Welling, Laura Thomson: PHP and MySQL Web Development, Pearson, 4th Edition, ISBN: 9788131729878

Suggested Additional Reading:

1. W. Jason Gilmore: Beginning PHP and MySQL 5 From Novice to Professional, Apress, ISBN: 9781590595527
2. Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass: Beginning PHP5, Apache, and MySQL Web Development, Wrox, ISBN: 9780764579660
3. Robin Nixon: Learning PHP, MySQL, and JavaScript, O'Reilly Media

Chapter wise Coverage from the main reference book(s):

Book # 1 : Chapters 1 – 13, 19,20, 21,23

Accomplishments of the student after completing the course:

Upon completion of the course, students will be able to efficiently continue to expand their web development knowledge on their own with the solid foundation gained in the course.

***** ***** *****