

# **JG UNIVERSITY**

## **SCHOOL OF COMPUTING**

State Private University Est. under the Gujarat Private University Act 2009

Drive-In Road, Ahmedabad, Gujarat - 380054

ASIA Charitable Trust



## **SYLLABUS**

### **Master of Computer Applications**

**2 years || 4 Semesters || Batch 2023 – 2025**

**Full-Time Programme**

**REGULATIONS – 2023**

# **JG UNIVERSITY**

## **SCHOOL OF COMPUTING**

### **Vision and Mission**

#### **Vision**

- To educate students with conceptual, practical, and research-oriented project development in the field of Information Technology meeting the needs of current and future industry demands to acquire global opportunities.

#### **Mission**

- To impart a strong foundation in conceptual, practical, and research-based focus to develop challenging projects on focusing analytical and problem-solving skills in the field of Information Technology.
- To ensure students with a higher employability ratio in the global industry, pursue higher studies and become successful entrepreneurs.
- To inculcate social, ethical, and moral values for the betterment of society

## JG UNIVERSITY

### SCHOOL OF COMPUTING

#### PEOs and POs

##### Program Educational Objectives (PEO)

- To enable graduates to be employed in the global Information Technology Industry.
- To ensure graduates will have their adaptability to work in existing technologies or shift to new technology in emerging technological changes in the growing industry.
- To motivate graduates to pursue higher education, take a step forward towards research, handle research-based projects, or become successful IT professionals or entrepreneurs.
- The approach of graduates will be professional and ethical in their work contributing to the society

##### Program Outcomes (PO)

**PO1: Domain Knowledge:** Apply the domain knowledge of Mathematics, Statistics, Computational fundamentals, and electives in Information Technology to solve complex problems in multi-disciplinary fields.

**PO2: Problem analysis:** Ability to identify, formulate, review literature, and analyze complex problems to derive solutions using domain knowledge.

**PO3: Design/development of solutions:** Ability to analyze the problem and find appropriate solutions for complex problems and design system components to meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems:** Use research-based ideas and research methodology to design experiments, analysis and interpretation of data, and create information to provide appropriate conclusions.

**PO5: Use of modern tools and technology:** Competent to use modern software tools and techniques with the acquired knowledge to create, select, and use appropriate techniques, resources, and modern IT tools for handling complex industrial projects.

**PO6: Ethics:** Inculcate professional ethics and values to be committed towards work and be a responsible citizen.

**PO7: Individual and teamwork:** Capable to work as an individual and in a team effectively for the role assigned as an individual, a member or leader in diverse teams, and a multidisciplinary environment.

**PO8: Communication:** Ability to communicate in a team effectively for managing complex projects and to comprehend and write effective reports, maintain proper documentation, make effective presentations, and give and receive clear instructions.

**PO9: Project management:** Transform acquired knowledge and understanding of domain knowledge, and apply them in projects, as a team member, and as a leader to manage projects in multidisciplinary environments.

**PO10: Life-long learning:** Identifying the objective and having the readiness to involve independent and life-long learning in the changing era of technological development.

## Program Structure

### Master of Computer Applications (MCA)

Programme Structure for Master of Computer Applications (MCA)						
Semester	Subject Head	Subjects				Total Credit
I	Core subjects	1. Object Oriented Programming with Java MCA230101(5) 2. Python Programming MCA230102(5) 3. Relational Database Management System MCA230103(5) 4. Networks & Cloud Infrastructure MCA230104(3)				24
	Foundation Course	5. Numerical and Statistical Methods MCA230105(3) 6. Professional Business Communication MCA230106(3)				
II	Core Subjects	1. Data Structures MCA230201(5) 2. Mobile Application Development MCA230202(5) 3. Elective - I MCA230203(5)				24
	Foundation Course	4. Software Engineering MCA230204(3) 5. Cyber Security MCA230205(3) 6. Software Testing and Quality Assurance MCA230206(3)				
	Elective Subject List	1. Angular Framework MCA230203A 2. Advanced Object Oriented Programming MCA230203B 3. Web Application Development Framework MCA230203C				
III	Core subjects	1. Machine Learning MCA230301(5) 2. Agile Computing MCA230302(3)				24
	Project Work	Mini Project MCA230303(8)				
	Elective Subject List	Elective - II Group A / B / C/D				
		Group A: Mobile Programming	Group B: Web Programming	Group C: Data Science	Group D: Internet of Things	
IV	Project Work	1. Mobile and Wireless Communication Networks MCA230304A1(3) 2. Cross Platform Application Development MCA230304A2(5)	1. Web Services and its Architecture MCA230304B1(3) 2. Open Source Framework MCA230304B2(5)	1. Data Science Essentials MCA230304C1(3) 2. Artificial Intelligence MCA230304C2(5)	1. Introduction to Internet of Things MCA230304D1(3) 2. Python for IoT MCA230304D2(5)	
		Industrial Project with Dissertation MCA230401				

## Curriculum Structure

**MASTER OF COMPUTER APPLICATIONS  
TWO YEARS POST GRADUATE PROGRAMME - EFFECTIVE FROM 2023-25  
REGULATIONS 2023**

<b>MCA Semester - I</b>										
<b>Subject Code</b>	<b>Subject title</b>	<b>Teaching Scheme (Per week)</b>				<b>Examination Scheme</b>				
		<b>Th</b>	<b>Tut</b>	<b>Pr</b>	<b>Total Credits</b>	<b>Internal</b>		<b>External</b>		<b>Total Marks</b>
						<b>Th</b>	<b>Pr</b>	<b>Th</b>	<b>Pr</b>	
MCA230101	Object Oriented Programming with Java	2	1	4	5	60	60	40	40	<b>200</b>
MCA230102	Python Programming	2	1	4	5	60	60	40	40	<b>200</b>
MCA230103	Relational Database Management System	2	1	4	5	60	60	40	40	<b>200</b>
MCA230104	Networks & Cloud Infrastructure	2	1	0	3	60	0	40	0	<b>100</b>
MCA230105	Numerical and Statistical Methods	2	1	0	3	60	0	40	0	<b>100</b>
MCA230106	Professional Business Communication	2	1	0	3	60	0	40	0	<b>100</b>
<b>Total Credits Earned</b>		<b>12</b>	<b>6</b>	<b>12</b>	<b>24</b>	<b>360</b>	<b>180</b>	<b>240</b>	<b>120</b>	<b>900</b>

<b>MCA Semester - II</b>										
<b>Subject Code</b>	<b>Subject title</b>	<b>Teaching Scheme (Per week)</b>				<b>Examination Scheme</b>				
		<b>Th</b>	<b>Tut</b>	<b>Pr</b>	<b>Total Credits</b>	<b>Internal</b>		<b>External</b>		<b>Total Marks</b>
						<b>Th</b>	<b>Pr</b>	<b>Th</b>	<b>Pr</b>	
MCA230201	Data Structures	2	1	4	5	60	60	40	40	<b>200</b>
MCA230202	Mobile Application Development	2	1	4	5	60	60	40	40	<b>200</b>
MCA230203	Elective - I	2	1	4	5	60	60	40	40	<b>200</b>
MCA230204	Software Engineering	2	1	0	3	60	0	40	0	<b>100</b>
MCA230205	Cyber Security	2	1	0	3	60	0	40	0	<b>100</b>
MCA230206	Software Testing & Quality Assurance	2	1	0	3	60	0	40	0	<b>100</b>
	<b>Elective - I Subject List</b>									
MCA230203A	Angular Framework									
MCA230203B	Advanced Object Oriented Programming									
MCA230203C	Web Application Development Framework									
<b>Total Credits Earned</b>		<b>12</b>	<b>6</b>	<b>12</b>	<b>24</b>					<b>900</b>

MCA Semester - III										
Subject Code	Subject title	Teaching Scheme (Per week)				Examination Scheme				
		Th	Tut	Pr	Total Credits	Internal		External		Total Marks
						Th	Pr	Th	Pr	
MCA230301	Machine Learning	2	1	4	5	60	60	40	40	200
MCA230302	Agile Computing	2	1	0	3	60	0	40	0	100
MCA230303	Mini Project	0	0	0	8	0	180	0	120	300
<b>Elective - II Group A: Mobile Programming</b>										
MCA230304A1	Mobile and Wireless Communication Networks	2	1	0	3	60	0	40	0	100
MCA230304A2	Cross Platform Application Development	2	1	4	5	60	60	40	40	200
<b>Elective - II Group B: Web Programming</b>										
MCA230304B1	Web Services and its Architecture	2	1	0	3	60	0	40	0	100
MCA230304B2	Open Source Framework	2	1	4	5	60	60	40	40	200
<b>Elective - II Group C: Data Science</b>										
MCA230304C1	Data Science Essentials	2	1	0	3	60	0	40	0	100
MCA230304C2	Artificial Intelligence	2	1	4	5	60	60	40	40	200
<b>Elective - Group D: Internet of Things</b>										
MCA230304D1	Introduction to Internet of Things	2	1	0	3	60	0	40	0	100
MCA230304D2	Python for IOT	2	1	4	5	60	60	40	40	200
<b>Total Credits Earned</b>		<b>8</b>	<b>4</b>	<b>8</b>	<b>24</b>					<b>900</b>

MCA Semester - IV										
Subject Code	Subject title	Teaching Scheme (Per week)				Examination Scheme				
		Th	Tut	Pr	Total Credits	Internal		External		Total Marks
						Th	Pr	Th	Pr	
MCA230401	Industrial Project with Dissertation	0	0	0	24	0	540	0	360	900
<b>Total Credits Earned</b>		<b>0</b>	<b>0</b>	<b>0</b>	<b>24</b>					<b>900</b>

**DETAILED SYLLABUS**

**FOR**

**MCA PROGRAMME**

**(1<sup>st</sup> SEMESTER)**



## MCA230101 – Object Oriented Programming with Java

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
5	2	1	4	60	60	40	40	200

### Prerequisites:

- Basics of programming and object-oriented concepts.

### Course Objectives:

- To understand the use of ODBC and JDBC
- To understand servlets, methods of servlets
- To understand servlet session tracking
- To understand JSP
- To understand struts architecture along with its strengths and weakness

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Database Connectivity</b> <ul style="list-style-type: none"> <li>Database Management</li> <li>ODBC API, JDBC API</li> <li>Querying a database, JDBC URL</li> <li>Driver Manager, Driver Connection</li> <li>Statement, ResultSet</li> <li>Configuring MYSQL and Oracle9i,</li> <li>Creating and Processing HTML Forms.</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Introduction To Java Servlets</b> <ul style="list-style-type: none"> <li>Webserver &amp; Introduction to Servlets</li> <li>Comparison between Servlets and Applets,</li> <li>Comparison between Servlets and other server-side technologies</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>Working of Servlets,</li> <li>Get and Post methods</li> <li>Javax servlet package</li> <li>Servlet Interface.</li> </ul> <p><b>HTTP Servlet</b></p> <ul style="list-style-type: none"> <li>Installing Tomcat on your machine</li> <li>HTTP Servlet,</li> <li>Need of HTTP Servlet class</li> <li>HTTP Request and HTTP Response</li> <li>Get and Post methods</li> <li>HTTP Servlet Request Interface</li> </ul>	
<b>Unit – 3</b>	<p><b>Working with Servlet Sessions</b></p> <ul style="list-style-type: none"> <li>Session tracking</li> <li>Techniques to keep track of sessions in servlets</li> <li>Cookies</li> </ul> <p><b>Inter Servlet Communication</b></p> <ul style="list-style-type: none"> <li>Session tracking</li> <li>Request Dispatcher Interface</li> <li>Method of Servlet Context Interface</li> <li>Method to get the object of Request Dispatcher</li> <li>Methods of Request Dispatcher interface</li> <li>Implementing Inter servlet communication via a problem statement</li> <li>Difference between forward() and sendRedirect() method</li> </ul>	<b>20%</b>
<b>Unit – 4</b>	<p><b>Java Server Pages</b></p> <ul style="list-style-type: none"> <li>Need for JSP</li> <li>Difference between Servlet and JSP</li> </ul> <p><b>Comment Tag and Scripting Element</b></p> <ul style="list-style-type: none"> <li>JSP Elements</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>Scripting Elements</li> <li>Implicit Objects</li> </ul> <b>JSP Directives</b> <ul style="list-style-type: none"> <li>Using the Core Tag Library</li> <li>Types of Directives</li> <li>JSP Directives</li> <li>Implicit Objects</li> <li>ErrorPage, Buffer</li> <li>Include directive, Taglib directive</li> <li>Using the Internationalization and Formatting Tag Library</li> </ul> <b>JSP Action Element and Custom Tags</b> <ul style="list-style-type: none"> <li>JSP Action Tags</li> <li>JSP Custom Tags</li> <li>Expression Language</li> <li>Model View Controller Architecture in JSP.</li> </ul>	
<b>Unit – 5</b>	<b>Exploring the Struts Architecture</b> <ul style="list-style-type: none"> <li>Why we need Struts?</li> <li>Why we need frameworks?</li> <li>Struts, Model 2 and MVC</li> <li>Struts Control Flow</li> <li>The strengths and weaknesses of Struts.</li> </ul> <b>Building a Simple Application</b> <ul style="list-style-type: none"> <li>Understanding @RequestMapping</li> <li>Strut by Strut</li> <li>Touring the logon application</li> <li>Dissecting the logon application</li> <li>Constructing an application.</li> </ul>	<b>20%</b>

### Learning Outcome:

This course covers:

LO1: Connecting with database with JDBC

LO2: Working with Servlets

LO3: Servlet with session tracking

LO4: Using JSP Directives

LO5: Using struts in an application

**Text Book:**

1. Java For Web Development, Sarika Agarwal and Vivek Gupta, BPB Publications
2. Practical Database Programming with Java, Ying Bai, Wiley-IEEE Press
3. Struts in Action, Building web applications with the leading Java framework, Ted N. Husted, Cedric Dumoulin, George Franciscus, David Winterfeldt, Manning Publications

**Reference Books & Web Links**

1. Java Servlet & JSP CookBook, Bruce W. Perry , O'Reilly.
2. J2EE: the complete reference, James Edward Keogh , McGraw-Hill
3. Java database programming bible, John O'Donahue, Wiley

**Unit wise coverage**

Unit 1: Chapter 1 of Text Book 1.

Unit 2: Chapter 3 and 4 of Text Book 1.

Unit 3: Chapter 5 and 6 of Text Book 1.

Unit 4: Chapter 7, 8, 9 and 10 of Text Book 1.

Unit 5: Chapter 2 and 3 of Text Book 3.

**List of Experiments**

1. Write down Five Basic steps to establish JDBC connection from Java Application. Also, mention sample code for each step.
2. Write a JDBC application which will interact with Database and perform the following task.
  - a. Create Student Table with RollNo, Name, and Address field and insert few records.
  - b. Using Statement Object, display the content of Record.
  - c. Using Statement Object, Insert Two Records.
  - d. Using Statement Object, Update One Record.
  - e. Using Statement Object, Delete One Record.
3. Design a JDBC application, which will demonstrate Scrollable ResultSet functionality.

4. Create Servlet for login page, if the username and password is correct then prints message “Hello username” else a message” login failed”.
5. Create Servlet that uses cookies to store the number of times a user has visited the servlet.
6. Write down the Program for testing the Servlet and study deployment descriptor.
7. Write down the program for testing the include action for servlet collaboration.
8. Write down the program for testing the forward action for servlet collaboration.
9. Create login form and perform state management using Cookies, Http Session and URL Rewriting.
10. Create Servlet file which contains the following functions:
  - a. Connect
  - b. Create Database
  - c. Create Table
  - d. Insert Records into respective table
  - e. Update records of particular table of database
  - f. Delete Records from table
  - g. Delete table and also database
11. Write down the Program which displays the simple JSP file.
12. Write down the program in which input the two numbers in an html file and then display the addition in JSP file.
13. Write down the program in which display the error by common file for all general pages.
14. Perform Database Access through JSP.
15. Write down the Program for testing the include action tag in jsp.
16. Write down the Program for testing the forward action tag.
17. Write down a program which demonstrates the core tag of JSTL.
18. Write down a program which demonstrates the Format tag of JSTL.
19. Write down a program which demonstrates the Function tag of JSTL
20. Write down a program which demonstrates the SQL tag of JSTL.

## MCA230102 – Python Programming

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
5	2	1	4	60	60	40	40	200

### Prerequisites:

- Knowledge of Object-Oriented Programming

### Course Objectives:

- To learn the fundamentals of Problem-Solving Techniques and Python programming basics.
- Studying detailed data representation, operator, control structures, and functions in Python and applying it to problems.
- To learn inbuilt functions in Python and apply it to mathematical problems.
- To learn advanced algorithms by implementing lists, tuples, and dictionaries.
- Integrating with database and to access the data stored in the database.

### Course Contents:

Units	Contents	Weightage
<b>Unit – 1</b>	<b>Introduction</b> <ul style="list-style-type: none"> <li>Introduction to Python</li> <li>Data types in Python</li> <li>Operators in Python: Membership Operators   Identity Operators</li> <li>Input and Output</li> <li>Control Statements</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Arrays, String, Functions Date and Time</b> <b>Arrays in Python</b> <ul style="list-style-type: none"> <li>Array   Advantages of Arrays   Creating an array   Importing an Array Module   Indexing and Slicing an Array   Processing the Arrays</li> </ul>	<b>20%</b>

	<p><b>String</b></p> <ul style="list-style-type: none"> <li>○ Creating String   Length   Indexing   Slicing   Repeating   Concatenation   Checking Membership   Comparing   Removing Space   Sub-string   String are Immutable   Replacing   Splitting and Joining   Sorting   Searching</li> </ul> <p><b>Date and Time</b></p> <ul style="list-style-type: none"> <li>○ Date and Time Now   Combining Date and Time   Formatting   Comparing   Sorting</li> </ul>	
<b>Unit – 3</b>	<p><b>Advanced Data types and Functions in Python</b></p> <ul style="list-style-type: none"> <li>• <b>List</b> <ul style="list-style-type: none"> <li>○ Exploring List   Creating lists using range() function   Updating the elements of the list   Concatenation of two lists   Repetition of lists   Membership in lists   Aliasing and Cloning lists   Methods to process List   Nested Lists</li> </ul> </li> <li>• <b>Tuple</b> <ul style="list-style-type: none"> <li>○ Tuples   Creating and accessing Tuple elements   Basic operations on Tuples   Functions to process tuples   Nested Tuples and sorting.</li> </ul> </li> <li>• <b>Dictionaries</b> <ul style="list-style-type: none"> <li>○ Introduction to Dictionaries   Operations on Dictionaries   Dictionary methods   Converting List into Dictionary   Passing dictionaries to functions.</li> </ul> </li> <li>• <b>Functions</b> <ul style="list-style-type: none"> <li>○ Difference between a function and a method   Defining-calling and returning (single and multiple) results from a function   Pass by Object Reference   Positional arguments   Keyword arguments   Default arguments   Variable length arguments   Anonymous Functions   Function Decorators.</li> </ul> </li> </ul>	<b>20%</b>
<b>Unit – 4</b>	<p><b>Classes, Inheritance, and Polymorphism</b></p> <ul style="list-style-type: none"> <li>• <b>Classes</b> <ul style="list-style-type: none"> <li>○ Creating a class   The Self variable   Constructor   Types of</li> </ul> </li> </ul>	<b>20%</b>

	<p>variables   Types of methods   Passing members of one class to another.</p> <ul style="list-style-type: none"> <li>• <b>Inheritance</b> <ul style="list-style-type: none"> <li>○ Implementing inheritance   Constructors in inheritance   Overriding Superclass constructors and methods   The super() method   Types of Inheritance</li> </ul> </li> <li>• <b>Polymorphism</b> <ul style="list-style-type: none"> <li>○ Introduction to polymorphism   Duck Typing Philosophy of Python   Operator overloading   Method overloading   Method overriding.</li> </ul> </li> </ul>	
<b>Unit – 5</b>	<p><b>Exception Handler</b></p> <ul style="list-style-type: none"> <li>○ Errors in a Python Program   Exceptions   Exception Handling   Types of Exceptions</li> </ul> <p><b>Database in Python</b></p> <ul style="list-style-type: none"> <li>• <b>Python Database Connectivity</b> <ul style="list-style-type: none"> <li>○ Types of Databases used in Python   Installation of MySQL Database Software   Working with MySQL Database   Using MySQL from Python   Retrieving all rows from a table   Inserting rows into a table   Deleting rows from a table   Updating rows in a table   Creating database tables through Python   Working with multiple table and accessing the data</li> </ul> </li> </ul>	<b>20%</b>

### Learning Outcome:

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

LO1: Apply to problem-solving techniques on real-world problems.

LO2: Implementing arrays, methods, and functions to perform reusability in programs

LO3: Applying advanced datatypes

LO4: Describing and applying object-oriented programming and database

### Text Book:

1. Core Python Programming, Dr. R. Nageswara Rao, Dreamtech Press



**Reference Books:**

1. Introduction to Programming Using Python, Y. Daniel Liang, Pearson Publications
2. Practical Programming - an Introduction to Computer Science Using Python by Jennifer Campbell, Paul Gries, Jason Montoyo, Greg Wilson.

**Chapter wise coverage :**

Unit 1: Chapter No. 1, 3, 4, 5, and 6 of Text Book

Unit 2: Chapter No. 7, 8 and 20 of Text Book

Unit 3: Chapter No. 9,10, and 11 of Text Book

Unit 4: Chapter No. 13 and 14 of Text Book

Unit 5: Chapter No. 16 and 24 of Textbook

**List of Experiments**

1. Write a program that evaluates an expression at run time using eval() function.
2. Write a menu driven python program which performs the following:
  - a. Find area of circle
  - b. Find area of triangle
  - c. Find area of square and rectangle
  - d. Find Simple Interest
3. Create a program name “employee.py” and implement the functions DA, HRA, PF, and ITAX. Create another program that uses the function of employee module and calculates gross and net salaries of an employee.
4. Create a dictionary that accepts cricket players’ names and scores in a match. Also we are retrieving runs by entering the player’s name.
5. Create a list and perform the following methods:

a. insert()	b. remove()	c. append()	
d. len()	e. pop()	f. clear()	g. delete()
6. Create a tuple and perform the following methods:
  - a. Add items()
  - b. Check item in a tuple()
  - c. len()
  - d. Access items()
7. Create a dictionary and apply the following methods:
  - a. Print the dictionary items
  - b. Access items
  - c. Use get()

- d. Change values
- e. Use len()
8. Write a program to create a menu with the following options:  
Accept user input and perform the operations. Use function with arguments
  - a. To perform addition
  - b. To perform subtraction
  - c. To perform multiplication
  - d. To perform division
9. Write a python program to print number is positive/negative using if-else statement
10. Write a python program for filter() to filter only even numbers from a given list.
11. Write a python program to print date, time for today and now
12. Write a python program to add some days to your present date and print the date added
13. Write a python program to count the number of characters in the string and store them in a dictionary data structure
14. Write a python program to count frequency of characters in a given file
15. Write a Python program to input Percentage. Calculate the percentage and grade according to the following:

Percentage	>=	90%	:	Grade	O
Percentage	>=	80%	:	Grade	A
Percentage	>=	70%	:	Grade	B
Percentage	>=	60%	:	Grade	C
Percentage	>=	50%	:	Grade	D
Percentage	>=	40%	:	Grade	E
Percentage	<	40%	:	Grade	F

16. Write a Python program to find the maximum between three numbers.
17. Write a program to input the basic salary of an employee and calculate its Gross salary according to the following:

Basic Salary	<=	10000	:	HRA	=	20%,	DA	=	80%
Basic Salary	<=	20000	:	HRA	=	25%,	DA	=	90%
Basic Salary	>	20000	:	HRA	=	30%,	DA	=	95%

**Looping Control**

18. Write a Python program to print the sum of the series  $1/2 + 1/3 + 1/4 + \dots + 1/N$ . Where N is natural number.
19. Write a Python program that prompts user to enter numbers. The process will repeat until user enters 0. Finally, the program prints sum of the numbers entered by the user.
20. Write a Python program to print all the numbers from 1 to 1000 that are not divisible by 2, 3, 5, 7, 11, 13, 17 and 19.
21. Write a Python program to find HCF (GCD) of two numbers.
22. Write a Python program to check whether a number is Armstrong number or not.
23. Write a Python program to swap first and last digits of a number.
24. Write a Python program for printing prime numbers up to N. ( $N > 100$ ).

**Functions**

25. Define a function to find the sum of all odd numbers between 1 to n.
26. Define a function to check whether a number is a palindrome or not.
27. Define a function to calculate the area of a circle using the formula.
28. Define a function to check whether number is perfect or not.
29. Define a function to print multiplication table of any number.
30. Define a function to print table of a number. Using this function display table of numbers from 1 to 10.
31. Define a recursive function to find power of a number.
32. Define a recursive function count number of digits in a number.
33. Write a recursive function to find a sum of  $15 + 25 + \dots + n5$ .
34. Write a python program to find the factorial value of a number using recursion.
35. Write a python program to implement Tower of Hanoi using recursive function.
36. Write function for finding factors (n) and use factors function to check whether given number n is prime or not.
37. Write a python program for printing Fibonacci series
38. Write recursive approach implementation
39. Write iterative implementation

**List, Tuples and Dictionary**

40. Write a Python program to create a list of each digit is a element in a list from a number. Example:

Input: 5467, Output: [5,4,6,7]

41. Write a Python program to find the second smallest number and second largest in a list.

42. Write a python program to create dictionary of index is the key and corresponding prime number as value up to 100. Output: {1:2, 2:3, 3:5, 4:7, 5:11, 6:13, 7:17, 8:19 and soon }

43. Write a Python program to find the smallest value and largest value in a dictionary.

Example: Input: D1={1:200,2:3000,3:100,5:20} output: 20, 3000.

44. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).

Sample Dictionary ( n = 5 ) :

Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

45. Write a Python program to convert a list of characters into a string. Example: Input: ['s','t','r','i','n','g'], Output: string.

46. Write a Python program to combine two dictionary adding values for common keys.

d1 = {'a': 10, 'b': 20, 'c':30}

d2 = {'a': 30, 'b': 20, 'd':40}

Sample output: {'a': 40, 'b': 40, 'd': 40, 'c': 30}

47. Write a program to print index at which a particular value exists. If the value exists a multiple location in the list, then print all the indices. Also, count the number of times the value is repeated in the list.

48. Write a program to remove all duplicate elements in a list.

49. Write a program to create a list of numbers in the range 1 to 10. Then delete all the odd numbers from the list and print the final list.

**Strings**

50. Write a program that counts up the number of vowels contained in the string S.

Valid vowels are: 'a', 'e', 'i', 'o', and 'u'.

For example, if s = 'azcbobobegghakl', your program should print: number of vowels 5

51. Assume s is a string of lower-case characters. Write a program that prints the number of times the string 'bob' occurs in s. For example, if s = 'azcbobobegghakl', then your program should print  
Number of times bob occurs is 2.
52. Write a Python program that finds whether a given character is present in a string or not. In case if it is present then it prints the index at which it is present. Do not use built-in find functions to search the character.
53. Write a Python program that counts the occurrence of a character in a string. Do not use built-in function.
54. Write a python program for following:
  - a. Take a input string with spaces, split it into list of words
  - b. From the list of words, create dictionary with keys (only unique words) and values (length of the word)
55. Write a python program to count number of vowels, spaces and to find longest word in a given input string. (Take input string with spaces)
56. Write a python program to reverse a string. Do not use inbuilt function.

### **Object Oriented Programming**

57. Write a Python program to create a student class (id, Name, mid1\_marks, mid2\_marks, quiz\_marks). Create a student objects and write a function marksList() to display student's result as given below:

ROLL NUMBER:

NAME:

MID1:

MID2:

QUIZ:

TOTAL: MID1+MID2+QUIZ

RESULT: A GRADE (IF TOTAL $\geq$ 80), B GRADE (TOTAL $<$ 80 and TOTAL $\geq$ 60), C GRADE (TOTAL $\geq$ 50 and TOTAL $<$ 60)

58. (Assume that maximum marks for mid\_term1 and mid2\_marks is 25 each , and quiz\_marks is 50).

59. Write a Python program to create a EMP class (id, Name, sal), create employee objects and write a function PaySlip(empobj) to display particular employee Pay Slip as given below:

EMP ID:

EMP NAME:

EMP BASIC: It is equal to sal. EMP HRA:

EMP DA:

EMP TAX:

EMP GROSS SAL: BASIC (sal) +HRA (18% of sal) +DA (10% of sal) EMP NET SAL: GROSS SAL-10% of GROSS SAL

60. Write a Python program to define a rectangle class with field's length and breadth. Define color rectangle class which is inherited from rectangle class with additional field color. Create N color rectangle objects and print which color rectangle is having minimum area.

61. Write a Python program to define CAR class (model, speed, price) and Firing CAR class which inherits from CAR with additional field number of bullets and fire method ().

62. Write a Program in python using object-oriented concept to create a base class called Polygon and there are three derived classes Named as triangle, rectangle and square.

63. The base class consists of the input function for accepting sides length

64. The derived classes must have output function for displaying area of triangle, rectangle and square.

## MCA230103 – Relational Database Management System

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
5	2	1	4	60	60	40	40	200

### Prerequisites:

- Knowledge of Database Management System

### Course Objectives:

- To practice designing, developing, and querying a database.
- To learn the fundamentals of data models conceptualize and implement in ER diagram.
- To study the principles to create an effective relational database model and write SQL queries to store, and retrieve data in different formats.
- To learn the concepts of transaction, concurrency control, and recovery procedures.
- To study internal storage, indexing techniques, query processing, and optimization.
- To learn the details of distributed, semi-structured, and unstructured data models.

### Course Contents:

Units	Contents	Weightage
Unit – 1	<b>Database Concepts &amp; Data Models</b> <ul style="list-style-type: none"> <li>• Database Systems <ul style="list-style-type: none"> <li>○ Why databases?   Data vs Information   Introducing the database   Database Systems</li> </ul> </li> <li>• Data Models <ul style="list-style-type: none"> <li>○ Data Modelling and Data Models   Importance of Data Models   Data Model basic Building Blocks   The Evolution of Data Models   Degrees of Data Abstraction</li> </ul> </li> </ul>	20%

Unit – 2	<b>The Relational Database Model, ERM and Normalization</b> <ul style="list-style-type: none"> <li>• The Relational database models <ul style="list-style-type: none"> <li>○ A Logical View   Keys   Integrity Rules  Relational set operators   The Data Dictionary and the system Catalog   Relationships within the relational model   Data Redundancy  Indexes  Codd’s Relational Rules</li> </ul> </li> <li>• The Entity-Relationship Model <ul style="list-style-type: none"> <li>○ The ERM   Normal forms   Developing an ER Diagram</li> </ul> </li> <li>• Normalisation of Database tables <ul style="list-style-type: none"> <li>○ Database Tables and normalization   The need for normalization   The Normalization Process   Higher-Level Normal Forms   Normalization and Database Design.</li> </ul> </li> </ul>	20%
Unit – 3	<b>Introduction to SQL</b> <ul style="list-style-type: none"> <li>• Data Definition Commands</li> <li>• Data Manipulation Commands <ul style="list-style-type: none"> <li>○ Saving Table Changes   Listing Table Rows   Updating Table Rows   Restoring Table Contents   Deleting Table Rows   Inserting Table Rows with a Select Subquery</li> </ul> </li> <li>• SELECT Queries <ul style="list-style-type: none"> <li>○ With Conditional Restrictions   With Logical Operators   With Special operators</li> </ul> </li> <li>• Additional DDL commands <ul style="list-style-type: none"> <li>○ Changing Colum’s Data Type   Changing Column’s Data Characteristic   Adding a column   Dropping a Column   Advanced Data Updates   Copying Parts of Tables   Adding Primary and Foreign Key Designations  Deleting a table</li> </ul> </li> <li>• Additional SELECT Query Keywords</li> <li>• Virtual Tables: Creating a View</li> <li>• Joining Database Tables</li> </ul>	20%



<b>Unit – 4</b>	<b>SQL Operators, Subqueries and Functions</b> <ul style="list-style-type: none"> <li>Relational SET Operators <ul style="list-style-type: none"> <li>UNION   UNION ALL   INTERSECT   MINUS</li> </ul> </li> <li>SQL Join Operators <ul style="list-style-type: none"> <li>Cross Join   Natural Join   Join USING Clause   Join ON Clause   Outer Joins</li> </ul> </li> <li>Subqueries and Correlated Queries <ul style="list-style-type: none"> <li>WHERE Subqueries   IN Subqueries   HAVING Subqueries   Multirow Subquery Operators: ANY and ALL   FROM Subqueries   Attribute List Subqueries   Correlated Subqueries</li> </ul> </li> <li>SQL functions <ul style="list-style-type: none"> <li>Date and Time Functions   Numeric Functions   String Functions   Conversion Functions</li> </ul> </li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Advanced SQL, Semi-structured &amp; Unstructured database</b> <ul style="list-style-type: none"> <li>Sequences</li> <li>Views</li> <li>Procedural SQL</li> <li>Overview of XML, DTD, XML schema, XML query languages, XML, XML, and databases</li> <li>Unstructured database: NoSQL an Overview</li> </ul>	<b>20%</b>

### Learning Outcome:

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

LO1: Design database schemas based on the conceptual model.

LO2: Formulate solutions to a query using relational algebra / SQL.

LO3: Understanding normalization and applying them in the database model.

LO4: Execute transactions and estimate the procedures Applying advanced datatypes

LO5: Describe distributed, semi-structured, and unstructured database systems.

LO6: Create XML database and validate with meta-data (XML schema).

LO7: To understand the concept of NoSQL

**Text Book:**

1. Database System: Design, Implementation and Management, Carlos Coronel, Steven Morris, Peter Rob, Course Technology, Cengage Learning, 9<sup>th</sup> Edition

**Reference Books:**

1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Seventh Edition, Pearson Education, 2017.
2. C. J. Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.
3. Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, Fourth Edition, Tata McGraw Hill, 2010.
4. G. K. Gupta, “Database Management Systems”, Tata McGraw Hill, 2011.
5. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Sixth Edition, Tata McGraw Hill, 2014.

**Chapter wise coverage:**

Unit 1: Chapter No. 1, and 2 of Text Book

Unit 2: Chapter No. 3, 4, and 6 of Text Book

Unit 3: Chapter No. 7 of Text Book

Unit 4: Chapter No. 8.1 to 8.4 of Text Book

Unit 5: Chapter No. 8.5 to 8.7 and 14.3 of Text Book

**List of Experiments****Implement the following concepts practically using MySQL/Oracle database**

1. Create a database table, add constraints (primary key, unique, check, Not null), insert rows, and update and delete rows using SQL DDL and DML commands.
2. Create a set of tables, add foreign key constraints, and incorporate referential integrity.
3. Query the database tables using different ‘where’ clause conditions and also implement aggregate functions.
4. Query the database tables and explore sub-queries and simple join operations.
5. Query the database tables and explore natural, equi and outer joins.
6. Write user-defined functions and stored procedures in SQL.
7. Execute complex transactions and realize DCL and TCL commands.

8. Write SQL Triggers for insert, delete, and update operations in a database table.
9. Create View and index for database tables with large number of records.
10. Create a XML database and validate it using XML schema.

### **Sample Queries**

#### **I. Consider the Company database with the following Schema**

EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, GENDER, SALARY,  
SUPERSSN, DNO) DEPARTMENT (DNAME, DNUMBER, MGRSSN, MSRSTARTDATE)

DEPT\_LOCATIONS (DNUMBER, DLOCATION)

PROJECT (PNAME, PNUMBER, PLOCATION, DNUM)

WORKS\_ON (ESSN, PNO<HOURS)

DEPENDENT (ESSN, DEPENDENT\_NAME, SEX, BDATE, RELATIONSHIP)

1. Perform the following:
  - a. Viewing all databases, Creating a Database, Viewing all Tables in a Database,
  - b. Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)
2. Perform the following:
  - a. Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / Restoring a Database.
3. For a given set of relation schemes, create tables and perform the following Simple Queries, Simple Queries with Aggregate functions, and Queries with Aggregate functions (group by and having clause).
4. Execute the following queries
  - a. How the resulting salaries if every employee working on the 'Research' Departments is given a 10% raise.
  - b. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department
5. Execute the following queries
  - a. Retrieve the name of each employee Controlled by department number 5 (use EXISTS operator).

- b. Retrieve the name of each dept and number of employees working in each department which has at least 2 employees
6. Execute the following queries
  - a. For each project, retrieve the project number, the project name, and the number of employee who work on that project.(use GROUP BY)
  - b. Retrieve the name of employees who born in the year
7. For each department that has more than five employees, retrieve the department number and number of employees who are making salary more than 40000.
8. For each project on which more than two employees work, retrieve the project number, project name and the number of employees who work on that project.
9. For a given set of relation tables perform the following
  - a. Creating Views (with and without check option), Dropping views, Selecting from a view

**II. Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries.**

BRANCH (Branchid, Branchname, HOD)

STUDENT (USN, Name, Address, Branchid, sem)

BOOK (Bookid, Bookname, Authorid, Publisher, Branchid)

AUTHOR (Authorid, Authurname, Country, age)

BORROW (USN, Bookid, Borrowed\_Date)

Perform the following:

- a. Viewing all databases, Creating a Database, Viewing all Tables in a Database, Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)

Execute the following Queries:

- b. List the details of Students who are all studying in 2nd sem MCA.
- c. List the students who are not borrowed any books.
- d. Display the USN, Student name, Branch\_name, Book\_name, Author\_name, Books\_Borrowed\_ Date of 2nd sem MCA Students who borrowed books.

- e. Display the number of books written by each Author.
- f. Display the student details who borrowed more than two books.
- g. Display the student details who borrowed books of more than one Author.
- h. Display the Book names in descending order of their names.
- i. List the details of students who borrowed the books which are all published by the same publisher.

**III. Consider the following schema:**

STUDENT (USN, name, date\_of\_birth, branch, mark1, mark2, mark3, total, GPA)

Perform the following:

- 1. Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit), and Undoing (rollback)

**Execute the following queries:**

- 2. Find the GPA score of all the students.
- 3. Find the students who born on a particular year of birth from the date\_of\_birth column.
- 4. List the students who are studying in a particular branch of study.
- 5. Find the maximum GPA score of the student branch-wise
- 6. Find the students whose name starts with the alphabet "S".
- 7. Update the column total by adding the columns mark1, mark2, mark3.
- 8. Find the students whose name ends with the alphabets "AR".
- 9. Delete the student details whose USN is given as 10

## MCA230104 Network & Cloud Infrastructure

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
3	2	1	0	60	0	40	0	100

### Prerequisites:

- Basic knowledge of Client and Server.

### Course Objectives:

- Learn the various cloud computing environments.
- Understand the core concepts of parallel, distributed computing and its architecture.
- Learn and implement virtualization environments.
- Insights of cloud computing architecture and its types.

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Introduction</b> <ul style="list-style-type: none"> <li>• Cloud computing at a glance <ul style="list-style-type: none"> <li>○ The vision of cloud computing</li> <li>○ Defining a cloud</li> <li>○ A closer look</li> <li>○ The cloud computing reference model</li> <li>○ Characteristics and benefits</li> <li>○ Challenges ahead</li> </ul> </li> <li>• Historical developments <ul style="list-style-type: none"> <li>○ Distributed systems</li> <li>○ Virtualization</li> <li>○ Web 2.0</li> <li>○ Service-oriented computing</li> </ul> </li> </ul>	<b>25%</b>
<b>Unit – 2</b>	<b>Principles of Parallel and Distributed Computing</b> <ul style="list-style-type: none"> <li>• Eras of computing</li> </ul>	<b>25%</b>

	<ul style="list-style-type: none"> <li>• Parallel vs. distributed computing</li> <li>• Elements of parallel computing <ul style="list-style-type: none"> <li>○ What is parallel processing?</li> <li>○ Hardware architectures for parallel processing</li> </ul> </li> <li>• Elements of distributed computing <ul style="list-style-type: none"> <li>○ General concepts and definitions</li> <li>○ Components of a distributed system</li> <li>○ Architectural styles for distributed computing</li> <li>○ Models for interprocess communication</li> </ul> </li> <li>• Technologies for distributed computing <ul style="list-style-type: none"> <li>○ Remote procedure call</li> <li>○ Distributed object frameworks</li> <li>○ Service-oriented computing</li> </ul> </li> </ul>	
<b>Unit – 3</b>	<b>Virtualization</b> <ul style="list-style-type: none"> <li>• Introduction</li> <li>• Characteristics of virtualized environments <ul style="list-style-type: none"> <li>○ Increased security</li> <li>○ Managed execution</li> <li>○ Portability</li> </ul> </li> <li>• Taxonomy of virtualization techniques <ul style="list-style-type: none"> <li>○ Execution virtualization</li> <li>○ Other types of virtualization</li> </ul> </li> <li>• Virtualization and cloud computing</li> <li>• Pros and cons of virtualization <ul style="list-style-type: none"> <li>○ Advantages of virtualization</li> <li>○ The other side of the coin: disadvantages</li> </ul> </li> <li>• Technology examples <ul style="list-style-type: none"> <li>○ VMware: full virtualization</li> </ul> </li> </ul> <b>Applications of Cloud</b> <ul style="list-style-type: none"> <li>• Scientific applications <ul style="list-style-type: none"> <li>○ Healthcare: ECG analysis in the cloud</li> <li>○ Biology: protein structure prediction</li> </ul> </li> </ul>	<b>25%</b>

	<ul style="list-style-type: none"> <li>○ Biology: gene expression data analysis for cancer diagnosis</li> <li>○ Geoscience: satellite image processing</li> <li>• Business and consumer applications <ul style="list-style-type: none"> <li>○ CRM and ERP</li> <li>○ Productivity</li> <li>○ Social networking</li> <li>○ Media applications</li> <li>○ Multiplayer online gaming</li> </ul> </li> </ul>	
<b>Unit – 4</b>	<b>Cloud Computing Architecture</b> <ul style="list-style-type: none"> <li>• Introduction</li> <li>• The cloud reference model <ul style="list-style-type: none"> <li>○ Architecture</li> <li>○ Infrastructure- and hardware-as-a-service</li> <li>○ Platform as a service</li> <li>○ Software as a service</li> </ul> </li> <li>• Types of clouds <ul style="list-style-type: none"> <li>○ Public clouds</li> <li>○ Private clouds</li> <li>○ Hybrid clouds</li> <li>○ Community clouds</li> </ul> </li> <li>• Economics of the cloud</li> <li>• Open challenges <ul style="list-style-type: none"> <li>○ Scalability and fault tolerance</li> <li>○ Security, trust, and privacy</li> <li>○ Organizational aspects</li> </ul> </li> </ul> <b>Cloud Platform in Industry</b> <ul style="list-style-type: none"> <li>• Amazon web services <ul style="list-style-type: none"> <li>○ Compute services</li> <li>○ Storage services</li> <li>○ Communication services</li> <li>○ Additional services</li> </ul> </li> <li>• Google AppEngine</li> </ul>	<b>25%</b>



	<ul style="list-style-type: none"> <li>○ Architecture and core concepts</li> <li>○ Application life cycle</li> <li>○ Cost model</li> <li>○ Observations</li> <li>● Microsoft Azure <ul style="list-style-type: none"> <li>○ Azure core concepts</li> <li>○ SQL Azure</li> <li>○ Windows Azure platform appliance</li> <li>○ Observations</li> </ul> </li> </ul>	
<b>Unit – 5</b>	<b>Case Study and Future of Cloud</b> <ul style="list-style-type: none"> <li>● Case Study</li> <li>● Dell, Wipro, Razorfish, and Japan Post</li> <li>● Future Trends in Cloud</li> <li>● Next Generation Networking</li> <li>● Mobile Cloud Architecture</li> <li>● Jungle Computing</li> </ul>	

### Learning Outcome:

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

LO1: How to design and implement the cloud computing paradigm

LO2: Understand theory and practical cloud computing development techniques

LO3: Design and implement cloud computing solutions

LO4: Focus on creating high-performance cloud applications

### Text Book:

1. Mastering Cloud Computing: Foundations and Applications Programming  
McGraw Hill Education By Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi

### Reference Book:

1. Essentials of Cloud Computing By K. Chandrasekaran Publication: CRC press

### Chapter wise coverage

Unit 1: Chapters No. 1 of Text Book

Unit 2: Chapters No. 2 of Text Book

Unit 3: Chapter No. 3, and 10 of Text Book

Unit 4: Chapter No. 4 and 9 of Text Book

Unit 5: Case Study

## MCA230105 – Numerical and Statistical Methods

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
3	2	1	0	60	0	40	0	100

### Prerequisites:

- Knowledge of basic Mathematics.

### Course Objectives:

- To solve linear and non-linear algebraic equations
- To perform operations of calculus, fit curves, and solve differential equations.
- To appreciate problems due to rounding errors and convergence.
- To develop familiarity with the different statistical methods used in problem-solving and decision-making.

### Course Contents:

Units	Content	Weightage
<b>Unit – 1</b>	<b>Computer Arithmetic &amp; Iterative Methods</b> <ul style="list-style-type: none"> <li>○ Floating Point representation of numbers</li> <li>○ Normalized floating point numbers</li> <li>○ Errors in numbers</li> <li>○ Solution of Linear and transcendental equations</li> <li>○ False Position</li> <li>○ Newton Raphson methods</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Interpolation and Approximation</b> <ul style="list-style-type: none"> <li>○ Lagrange's interpolation</li> <li>○ Forward difference</li> <li>○ Backward difference</li> <li>○ Inverse interpolation</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>○ Linear Regression and Non-Linear Regression (Least square Curve fitting)</li> <li>○ Numerical Differentiation: Newton's forward and backward difference formulae</li> </ul>	
<b>Unit – 3</b>	<b>Solution of Simultaneous Equations &amp; Ordinary Differential Equations</b> <ul style="list-style-type: none"> <li>○ Gauss Elimination method</li> <li>○ Gauss Seidal iterative method</li> <li>○ Euler's Method</li> <li>○ Runge-Kutta second order method.</li> <li>○ Numerical Integration: Concept of Numerical Integration by Simpson's 1/3 rule &amp; 3/8 rules</li> </ul>	<b>20%</b>
<b>Unit – 4</b>	<b>Measures of Central tendency, dispersion and Probability</b> <ul style="list-style-type: none"> <li>○ Introduction to measures of central tendency - mean, median, mode Measures of dispersion, Standard Deviation</li> <li>○ Probability, addition rule, mutually exclusive events, multiplication rule, probability under statistical independence, probability under statistical dependence, and conditional probability.</li> <li>○ Probability distributions-binomial, Poisson, and normal distribution</li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Statistical inference theory</b> <ul style="list-style-type: none"> <li>○ Sample distributions</li> <li>○ Testing of hypothesis</li> <li>○ One tail and two tail tests</li> <li>○ Tests of significance (about mean)</li> <li>○ Parametric &amp; non-parametric tests</li> <li>○ Tests of Significance: Chi square test of independence, t test</li> </ul>	<b>20%</b>

### Learning Outcome:

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

LO1: Get acquainted with the different numerical methods used in problem solving.

LO2: Logical understanding through the concepts learned in the class, which is the base of computer science.

LO3: Get acquainted with essential ideas and reasoning of applied statistics like data analysis, distributions and inference theory.

LO4: Learn a statistical technique through different tools and apply it to case studies using the concepts learned in the class

**Text Book:**

1. “Numerical Methods” – E. Balaguruswamy (TMH publications)
2. Srimanta Pal, “Numerical Methods”, Oxford University Press
3. Richard Levin, David Rubin, “Statistics for Management”, 7th edition, PHI
4. Anderson Sweeney Williams, “Statistics for Business and Economics”, 11<sup>th</sup> edition

**Reference Books:**

1. Steven C Chapra, Raymond P Canale, “Numerical Methods for Engineers”, 5th Edition, Tata McGraw Hill Publication, Special Indian Edition
2. “Computer Oriented Numerical Methods”- Dr B.S Grewal
3. “Numerical Methods – Problems and Solutions” – M. K. Jain and R. K. Jain
4. “Computer Oriented Numerical Methods” – V. Rajaraman (PHI publications)
5. S.P Gupta, “Statistical Methods”, Himalaya Publication

## MCA230106 – Professional Business Communication

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
3	2	1	0	60	0	40	0	100

### Prerequisites:

- Quest to develop strong foundation for communication
- Understanding the importance of communication in business
- Knowledge to articulate verbal and written communication

### Course Objectives:

- To introduce business communication to students
- To comprehend network communication's various facets.
- To become familiar with modern developments in communication technology and how they are used in business.
- To educate them on the value of formal business communication and protocol.

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Language &amp; Communication</b> <ul style="list-style-type: none"> <li>• Sentence Construction &amp; Punctuations</li> <li>• Concept of Business Communication</li> <li>• Art of Summarizing</li> <li>• Listening Skills</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Business Ethics</b> <ul style="list-style-type: none"> <li>• Managing Organisation Structure</li> <li>• Making Ethical Decisions</li> <li>• Characteristics of Business Ethics</li> <li>• Importance of Business Ethics</li> </ul>	<b>20%</b>

<b>Unit – 3</b>	<b>Writing Skills</b> <ul style="list-style-type: none"><li>• Report Writing &amp; Business Letters</li><li>• Note &amp; Memo Writing</li><li>• Analytical and project report</li><li>• CV, cover letter, personal statement</li></ul>	<b>20%</b>
<b>Unit – 4</b>	<b>Speaking Skills</b> <ul style="list-style-type: none"><li>• Extempore Speech</li><li>• Business presentations with multimedia</li><li>• Group Discussions</li><li>• Interview Skills</li></ul>	<b>20%</b>
<b>Unit - 5</b>	<b>Business Communication</b> <ul style="list-style-type: none"><li>• Business Correspondence</li><li>• Professional Etiquettes</li><li>• Professional Grooming</li><li>• Practical</li></ul>	<b>20%</b>

**Learning Outcome:**

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

LOL1: Develop and expand their understanding about business communication

LOL2: Comprehend the different facets of business communication

LOL3: Convey speech through clarity

**Textbooks :**

1. Business Communication for Success, UNIVERSITY OF MINNESOTA LIBRARIES PUBLISHING EDITION, 2015.
2. Crucial Conversations, Kerry Patterson, Joseph Grenny, Ron Macmillan
3. Connecting with Others, John C. Maxwell
4. Introduction to Business Communication (online PDF)

**Reference Books & Web Links:**

1. Meenakshi Raman and Sangeetha Sharma. 2012. Technical Communication. New Delhi: OUP
2. Rizvi, M. A. 2005. Effective Technical Communication. New Delhi: Tata McGraw Hill
3. Sanjay Kumar and Pushp Latha. 2012. Communication Skills. New Delhi: OUP
4. Er. A. K. Jain, Dr. Pravin S. R. Bhatia and Dr. A. M. Sheikh. 2013. Professional Communication Skills. S. Chand Publishers. New Delhi.
5. Farhathullah, T.M. 2009. English for Business Communication. Bangalore: Prism Publishers
6. Bikram K Das. 2011. Functional Grammar and Spoken and Written Communication in English. Kolkata: Orient Blackswan
7. Kiranmai Dutt, P et al. 2011. A Course in Communication Skills. New Delhi: CUP India
8. Krishnaswamy, N. 2000. Modern English – A Book of Grammar, Vocabulary and Usage. Macmillan India Pvt. Ltd
9. Ramachandran, K K. et al. 2007. Business Communication. New Delhi: Macmillan
10. Taylor, Ken. 2011. 50 ways to improve your Business English. Hyderabad: Orient Blackswan
11. Basics of Business Communication – by Lesikar & Flatley – Tata McGraw Hill
12. Business Correspondence & Report Writing, Sharma, TMH
13. Business Communication Strategies, Monipally, TMH
14. English for Technical communication, Laxminarayanan, Scitech
15. Business Communication, Kaul, PH
16. Communication Skill for Effective Mgmt., Ghanekar, EPH
17. [https://www.icsi.edu/media/webmodules/CSEET/BUSINESS\\_COMMUNICATION\\_printable.pdf](https://www.icsi.edu/media/webmodules/CSEET/BUSINESS_COMMUNICATION_printable.pdf)
18. <https://www.investopedia.com/terms/b/business-ethics.asp#:~:text=Business%20ethics%20refers%20to%20implementing,social%20responsibility%2C%20and%20fiduciary%20responsibilities.>
19. <https://www.wallstreetmojo.com/business-ethics/>
20. <https://www.onlinemanipal.com/blogs/what-is-business-communication#:~:text=Business%20communication%20is%20exchanging%20information,errors%20and%20enhance%20organizational%20procedures.>



**DETAILED SYLLABUS  
FOR  
MCA PROGRAMME  
(2<sup>nd</sup> SEMESTER)**

## MCA230201 – Data Structures

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
5	2	1	4	60	60	40	40	200

### Prerequisites:

- Knowledge of C Programming language with controls, and algorithm development.

### Course Objectives:

The objective of the course is to,

- Introduce the fundamentals of Data Structures, Abstract concepts
- Analyse step-by-step and develop algorithms to solve real-world problems.
- Implementing Stacks, Queues, Linked Lists, Trees and Graphs.
- Implementing various searching & sorting techniques

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Introduction to Data Structure, Linked List, and Stacks</b> <ul style="list-style-type: none"> <li>Introduction <ul style="list-style-type: none"> <li>Classification of Data Structures</li> <li>Operations on Data Structures</li> <li>Abstract Data Type</li> <li>Different Approaches to Designing an Algorithm</li> </ul> </li> <li>Linked Lists <ul style="list-style-type: none"> <li>Introduction <ul style="list-style-type: none"> <li>Basic terminologies</li> <li>Memory Allocation and De-allocation for a linked list.</li> </ul> </li> </ul> </li> <li>Singly Linked Lists, Circular Linked Lists, Doubly Linked Lists, <ul style="list-style-type: none"> <li>Traversing, Insertion, and Deletion operations on linked list</li> </ul> </li> <li>Applications of Linked Lists</li> </ul>	<b>20%</b>

<b>Unit – 2</b>	<b>Stack &amp; Queues</b> <ul style="list-style-type: none"> <li>• Stacks <ul style="list-style-type: none"> <li>○ Introduction</li> <li>○ Array Representation of Stacks</li> <li>○ Operations on Stack</li> <li>○ Push, Pop and Peek Operations</li> <li>○ Linked Representation of Stacks</li> <li>○ Operations on a Linked Stack <ul style="list-style-type: none"> <li>▪ Push, Pop Operations</li> </ul> </li> <li>○ Applications of Stacks <ul style="list-style-type: none"> <li>▪ Reversing a List</li> <li>▪ Implementing Parentheses Checker</li> <li>▪ Evaluation of Arithmetic Expressions</li> <li>▪ Recursion</li> </ul> </li> </ul> </li> <li>• Queues <ul style="list-style-type: none"> <li>○ Introduction to Queues</li> <li>○ Array Representation of Queues</li> <li>○ Linked Representation of Queues</li> <li>○ Types of Queues <ul style="list-style-type: none"> <li>▪ Circular Queues   Deques   Priority Queues   Multiple Queues</li> </ul> </li> <li>○ Applications of Queues</li> </ul> </li> </ul>	<b>20%</b>
<b>Unit – 3</b>	<b>Trees</b> <ul style="list-style-type: none"> <li>○ Introduction <ul style="list-style-type: none"> <li>▪ Basic Terminology</li> </ul> </li> <li>○ Types of Trees <ul style="list-style-type: none"> <li>▪ General Trees</li> <li>▪ Forests</li> <li>▪ Binary Trees</li> <li>▪ Binary Search Trees</li> <li>▪ Expression Trees</li> </ul> </li> <li>○ Creating a Binary Tree from a General Tree</li> <li>○ Traversing a Binary Tree</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>▪ Pre-order Traversal</li> <li>▪ In-order Traversal</li> <li>▪ Post-order Traversal</li> <li>○ Binary Search Trees <ul style="list-style-type: none"> <li>▪ Operations on Binary Search Trees <ul style="list-style-type: none"> <li>▪ Searching for a Node in a Binary Search Tree</li> <li>▪ Inserting a New Node in a Binary Search Tree</li> <li>▪ Deleting a Node from a Binary Search Tree</li> </ul> </li> <li>▪ Threaded Binary Trees <ul style="list-style-type: none"> <li>▪ Traversing a Threaded Binary Tree</li> </ul> </li> <li>▪ Introduction to AVL Trees</li> </ul> </li> <li>○ Multi-way Search Trees <ul style="list-style-type: none"> <li>▪ Introduction to B Trees</li> </ul> </li> </ul>	
<b>Unit – 4</b>	<b>Heaps and Graphs</b> Heaps <ul style="list-style-type: none"> <li>• Binary Heaps <ul style="list-style-type: none"> <li>○ Inserting a New Element in a Binary Heap</li> <li>○ Deleting an Element from a Binary Heap</li> <li>○ Applications of Binary Heaps</li> </ul> </li> <li>• Introduction to Binomial Heaps, Fibonacci Heaps</li> <li>• Comparison of Binary, Binomial and Fibonacci Heaps</li> <li>• Applications of Heaps</li> </ul> Graphs <ul style="list-style-type: none"> <li>• Introduction</li> <li>• Graph Terminology</li> <li>• Directed Graphs <ul style="list-style-type: none"> <li>○ Terminology of a Directed Graph</li> <li>○ Transitive Closure of a Directed Graph</li> </ul> </li> <li>• Representation of Graphs <ul style="list-style-type: none"> <li>○ Adjacency Matrix Representation</li> <li>○ Adjacency List Representation</li> </ul> </li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>○ Adjacency Multi-List Representation</li> <li>● Graph Traversal Algorithms <ul style="list-style-type: none"> <li>○ Breadth-First Search Algorithm</li> <li>○ Depth-first Search Algorithm Topological Sorting</li> </ul> </li> <li>● Shortest Path Algorithms <ul style="list-style-type: none"> <li>○ Minimum Spanning Trees</li> <li>○ Prim's Algorithm</li> <li>○ Kruskal's Algorithm</li> <li>○ Dijkstra's Algorithm</li> </ul> </li> <li>● Applications of Graphs</li> </ul>	
<b>Unit – 5</b>	<b>Searching and Sorting</b> <ul style="list-style-type: none"> <li>● Introduction to Searching <ul style="list-style-type: none"> <li>○ Linear Search</li> <li>○ Binary Search</li> </ul> </li> <li>● Introduction to Sorting <ul style="list-style-type: none"> <li>○ Bubble Sort</li> <li>○ Insertion Sort</li> <li>○ Selection Sort</li> <li>○ Merge Sort</li> <li>○ Quick Sort</li> </ul> </li> </ul>	<b>20%</b>

### Learning Outcome:

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

LO1: To understand the concepts of data structures and to solve the real-world problem

LO2: Can able to handle sorting, searching, insertion, deletion, and traversing on various data structures.

LO3: Familiar with basic algorithm analysis and their implementation.

### Text Book:

1. Data Structure Using C, Reema Thareja, Oxford University Press

### Reference Books:

1. An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill.

2. Data Structures using C & C++ -By Ten Baum Publisher – Prentice-Hall International.
3. Fundamentals of Computer Algorithms by Horowitz, Sahni, Galgotia Pub. 2001.

**Chapter wise coverage :**

Unit 1: Chapter No. 2, 6 of Textbook

Unit 2: Chapter No. 7,8 of Textbook

Unit 3: Chapter No. 9,10, 11 of Textbook

Unit 4: Chapter No. 12,13 of Textbook

Unit 5: Chapter No. 14 of Textbook

**List of Experiments****Linked List**

1. Implement the operations on a singly linked list, doubly linked list, and circular linked list.

**Stack**

2. Create a Stack and do the following operations using arrays and linked lists
  - (i) Push
  - (ii) Pop
  - (iii) Peep

**Queue**

3. Create a Queue and do the following operations using arrays and linked lists
  - (i) Add
  - (ii) Remove
4. Circular Queue implementation using array & linked list

**Graph**

5. Perform the following operations in a given graph
  - (i) Depth first search
  - (ii) Breadth first search

**Tree**

6. Tree traversal using recursive and non-recursive
7. Create a binary search tree and do the following traversals
  - (i) In-order
  - (ii) Pre order
  - (iii) Post order
8. Implement the following operations on a binary search tree.
  - (i) Insert a node
  - (ii) Delete a node

**Searching**

9. Write a program to search an element in an array using the linear search technique.
10. Write a program to search an element in an array using binary search.

**Sorting**

11. Write a program to enter n numbers in an array. Redisplay the array with elements being sorted in ascending order.
12. Write a program to sort an array using the insertion sort algorithm.
13. Write a program to sort an array using the selection sort algorithm.
14. Write a program to implement merge sort.

## MCA230202 – Mobile Application Development

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
5	2	1	4	60	60	40	40	200

### Prerequisites:

- Basic knowledge of Java Programming language.

### Course Objectives:

The objective of the course is to,

- Introduce the lifecycle of Mobile App Development.
- To explore the various prototypes for hybrid and native mobile application
- To expertise in software development methodologies for deploying mobile applications

### Course Contents:

Units	Content	Weightage
<b>Unit – 1</b>	<b>Android Fundamentals</b> <ul style="list-style-type: none"> <li>• Android Fundamentals-Getting Started with Android</li> <li>• Mastering the Android Development Tools</li> <li>• Building Android Applications</li> <li>• Installing Eclipse IDE and Android SDK</li> <li>• Configuring Development Hardware</li> <li>• Managing Application Resources</li> <li>• Configuring the Android Manifest File</li> <li>• Designing an Application Framework</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Building an Application Framework</b> <ul style="list-style-type: none"> <li>• Implementing an Animated Splash Screen</li> <li>• Implementing the Main Menu Screen</li> </ul>	<b>20%</b>



	<ul style="list-style-type: none"> <li>Developing the Help and Scores Screens.</li> <li>Building Forms to Collect User Input</li> <li>Using Dialogs to Collect User Input</li> <li>Adding Application Logic</li> </ul>	
<b>Unit – 3</b>	<b>Android Features</b> <ul style="list-style-type: none"> <li>Working with Images and the Camera</li> <li>Adding Support for Location-Based Services</li> <li>Adding Network Support</li> <li>Adding Social Features.</li> </ul>	<b>20%</b>
<b>Unit – 4</b>	<b>Using Android APIs</b> <ul style="list-style-type: none"> <li>Using Android Data and Storage APIs</li> <li>Managing data using SQLite</li> <li>Sharing Data between Applications with Content Providers</li> <li>Using Android Networking APIs</li> <li>Using Android Web APIs</li> <li>Using Android Telephony APIs</li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Internationalizing and Testing Android App</b> <ul style="list-style-type: none"> <li>Internationalizing Your Application</li> <li>Developing for Different Devices</li> <li>Testing Android Application</li> </ul> <b>Publishing Android Application</b> <ul style="list-style-type: none"> <li>Getting Ready to Publish</li> <li>Publishing on the Android Market</li> </ul>	<b>20%</b>

### Learning Outcome:

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

LO1: Design and develop a mobile application using Android SDK

LO2: Competent with characterization and architecture of mobile applications.

LO3: Understanding various engineering works involved in mobile development.

**Text Book:**

1. Lauren Darcey, Shane Conder Teach Yourself Android Application Development in 24 Hours, 2014, Third edition, Sams Publishing
2. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, 2nd edition, Pearson Education

**Reference Books :**

1. Wei-Meng Lee, Beginning Android 4 Application Development, 2012, 1st Edition, John Wiley & Sons.
2. Reto Meier, Professional Android 4 Application Development, 2012, Third Edition, Wrox.

**Chapter wise coverage :**

Unit 1: Part - 1 of Text Book-1

Unit 2: Part - 2 of Text Book-1

Unit 3: Part - 3 of Text Book-1

Unit 4: Chapters 10, 11, 12, 13, 16 of Text Book-2

Unit 5: Part – 4, 5 of Text Book-1

**List of Experiments**

1. Print “Welcome to Mobile App Development” in the middle of the screen in various styles.
2. Create a sample application with a login module. (Check username and password)
  - a. On successful login, go to the next screen.
  - b. And on failing login, alert user using Toast.
  - c. Also pass username to next screen
  - d. Till the login is validated the button should remain disabled
3. Create a UI such that, one screen has a list of all the types of cars. On selecting of any car name, the next screen should show Car details like: name, launch date, and company name.
4. Create a Registration page to demonstrate Basic widgets available in Android.
5. Write an Android app to get the current location using GPS.
6. Create an application that will Demonstrate Dialog Box Control In Android
7. Create an application for demonstration of Scroll view in Android.
8. Create an application for demonstration of Relative and Table Layout in Android.

9. Write an Android program to display stationery items in the Main Activity with the check box. Select the items and generate the bill. Include VAT as a toggle button, to calculate the bill. For members/ Non-members use the radio button and give a 2% discount on the bill amount
10. Create a SQLite database that contains an EMPLOYEE table. The EMPLOYEE table contains the Emp.no, Name, and Basic Salary. Do the following operations by clicking the respective button.
  - a) Add – Insert a new record.
  - b) Delete – Delete the record with the given Emp. No.
  - c) VIEW - To display the details of the employee for the given number.Calculate gross salary and display it
11. Understanding content providers and permissions:
  - Read phonebook contacts using content providers and display in list
12. Write an Android app to give a Notification Course Registration form for multiple student registrations using Fragments
13. Write an Android app to pass information in bundles and reply to the result back on the same page
14. Date Picker Dialog: Illustrate the DatePickerDialog application as described here.

On the launch of Emulator, it will display the following Screen (1). Now you can see that the date has already been set at the bottom label. Now we will change the date through DatePickerDialog by pressing the Set Date button. On pressing the button following Screen (2) would appear. Now set the required date, and after setting the date, press the Done button. This dialog will disappear and your newly set date will start showing on the Screen (3).
15. Time Picker Dialog: Illustrate the TimePickerDialog application as described here. On the launch of Emulator, it will display the following Screen (1). Now you can see that the time has already been set for the TimePicker widget. And the current time is also shown at the bottom label. Now we will change the time and press the save button. As you can see in Screen(2), that the time has been updated after pressing the save button.

## MCA230203A – Angular Framework

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
5	2	1	4	60	60	40	40	200

### Prerequisites:

- Basic knowledge of JavaScript

### Course Objectives:

- Understand the features of JavaScript
- Allows learning the Basics of AngularJS
- Deep understanding of MVC architecture and high-level application design.
- Allows implementing directives, filters, and routing system in AngularJS
- Enables to work with forms, views, and AngularJS Animation

### Course Contents:

Units	Contents	Weightage
<b>Unit – 1</b>	<b>JavaScript You Need to Know</b> <ul style="list-style-type: none"> <li>• JavaScript Primer</li> <li>• Working with Objects</li> </ul> <b>Basics of AngularJS</b> <ul style="list-style-type: none"> <li>• Why we need Frameworks</li> <li>• Downloading and Installing AngularJS</li> <li>• Browser Support</li> <li>• Your First AngularJS Application</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Introduction to MVC</b> <ul style="list-style-type: none"> <li>• Design Patterns</li> </ul> <b>Filters and Modules</b> <ul style="list-style-type: none"> <li>• Introduction to Filters</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>Built-in Filters</li> <li>Angular JS Modules</li> <li>Bootstrapping AngularJS</li> <li>Creating a Custom Filter</li> </ul> <b>Directives</b> <ul style="list-style-type: none"> <li>The Basics of Directives</li> <li>Using Directives</li> <li>Built-in Directives</li> <li>Creating a Custom Directive</li> </ul>	
<b>Unit – 3</b>	<b>Working with Forms</b> <ul style="list-style-type: none"> <li>HTML Forms Overview</li> <li>AngularJS Forms</li> <li>Validating Forms</li> </ul> <b>Services and Server Communication</b> <ul style="list-style-type: none"> <li>Using Services</li> <li>Creating Services</li> <li>Server Communication</li> </ul>	<b>20%</b>
<b>Unit – 4</b>	<b>Organizing Views</b> <ul style="list-style-type: none"> <li>Installing the ngRoute Module</li> <li>Using URL Routes</li> </ul> <b>AngularJS Animation</b> <ul style="list-style-type: none"> <li>Installing the ngAnimate Module</li> <li>CSS Animation Overview</li> <li>Applying Animations</li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Deployment Considerations</b> <ul style="list-style-type: none"> <li>Configuration</li> <li>Testing</li> <li>Error Handling</li> <li>Hide Unprocessed Templates</li> <li>Minification and Bundling</li> <li>Managing the Build Process</li> </ul>	<b>20%</b>

**Learning Outcome:**

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

LO1: Understanding of JavaScript

LO2: To understand MVC architecture and high level application design.

LO3: Understanding directives, filters, routing system in AngularJS

**Text Book:**

1. Beginning Angular JS, Andrew Grant, Apress

**Reference Books & Web Links:**

2. Angular Up & Running, Learning Angular Step by Step, Shyam Seshadri, O'reilly
3. Learning AngularJS: A Guide to AngularJS Development, Ken Williamson, O'reilly
4. <https://docs.AngularJS.org/guide>

**Chapter wise coverage :**

Unit 1: Chapter No. 1, 2 of Text Book

Unit 2: Chapter No.3, 4 and 5 of Text Book

Unit 3: Chapter No. 6, 7 of Text Book

Unit 4: Chapter No. 8, 9 of Text Book

Unit 5: Chapter No. 10 of Text Book

**List of Experiments****Programming using AngularJS**

1. Create first AngularJS application.
2. Print live updates.
3. To add two numbers.
4. Initialize two variables using ng-init and print the result.
5. To concatenate to Strings.
6. To display array data using Expression.
7. To print the product of 2 numbers.
8. To perform the Arithmetic operation on two numbers.

9. Write an AngularJS script to display the Student name, Roll no and calculate the Percentage (use ng-init to initialize name, roll and all subject marks)
10. Write an AngularJS script to display the list of games stored in an array on click of button using ng-click. And also demonstrate ng-init, ng-binding directive of Angular js?
11. Write an AngularJS script for addition of two numbers using ng-init, ng-model & ng-bind. And also Demonstrate ng-show, ng-disabled, ng-click directives on the button component.
12. Write an AngularJS script to display the Product name, Quantity, Rate, and Total Price (use ng-init to initialize values)
13. Write Angular JS by using ng-click Directive to display an alert message after clicking the element.
14. Using Angular js, display the 10 student details in Table format (using ng-repeat directive use Array to store data)
15. Write an AngularJS script to print details of the bank (bank name, MICR code, IFC code, address, etc.) in tabular form using ng-repeat.
16. Write an HTML code using Angular JS to generate the following output (hint: use ng-repeat, ng-init directive)

#### Undergraduate Courses

- i. BBA
- ii. BCA
- iii. B.Sc.(IT)
- iv. B.Com

#### Post Graduate Courses

- i. M.Sc (IT)
- ii. MCA

17. Using AngularJS display the student information (use CSS)
18. Create a Simple AngularJS Script to print the Student Marks sheet with grade?
19. To accept the details of staff having field Id, name, address, and salary. Display the details in a table format. (use MVC.)
20. To print e-ticket for the ticket booking which has name, address, contact no., gender, Date of booking, date of journey, name of passengers, etc.
21. To show the syllabus content of all subjects of FYBCA (use ng-view)

**Form Validation**

22. Create the student registration form and validate all the fields?
23. Store 5 students' information in an array and display students' information in table form orderBy Name (use orderBy filter sorts an array)
24. To display the student details who live in Ahmedabad in Table format (using ng-repeat directive, use Array to store data, use filter)
25. To search a product with its rate (use ng-repeat directive, use Array to store data, use filter currency)
26. To search student names according to the keyword typed and display details (use array and filter).
27. To display the Employee details order by salary in Table format (using ng-repeat directive, use Array to store data, use filter)
28. Using AngularJS to accept the details such as name, mobile number, pincode, and email address and make validation.
  - a. Name should contain character only
  - b. Mobile number should contain only 10 digits
  - c. The pincode should contain only 6 digits
  - d. email id should contain only one @ symbol
29. Create a Login System.



## MCA230203C - Web Application Development Framework

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
5	2	1	4	60	60	40	40	200

### Prerequisites:

- Knowledge of Client Server Architecture, Use of Controls & Server, database, ASP.Net web services

### Course Objectives:

- To develop basic knowledge of designing and developing client server architecture-based web applications using Asp.Net with C#
- This course gives an understanding of MVC 5.0 architecture and implementation of MVC architecture in entity framework.
- This course covers the advanced topic in ASP.NET with JSON, AJAX, JQUERY and Web API so that students can develop any web-based advanced projects for the industry.

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Introduction to Asp.Net Basics</b> <ul style="list-style-type: none"> <li>• Introduction to .NET Framework: <ul style="list-style-type: none"> <li>○ NET framework, Namespaces, Assemblies.</li> </ul> </li> <li>• Building an ASP.NET Web Site: <ul style="list-style-type: none"> <li>○ Creating Web Sites, Working with Files in Your Web Site, Working with Web Forms.</li> </ul> </li> <li>• <b>Introduction to Server Controls:</b> <ul style="list-style-type: none"> <li>○ A Closer Look at ASP.NET Server Controls, Types of Controls, ASP.NET State Engine</li> </ul> </li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>• <b>Introduction to Programming:</b> <ul style="list-style-type: none"> <li>○ Data Types and Variables, Statements, Organizing Code.</li> </ul> </li> <li>• <b>Introduction to Databases</b> <ul style="list-style-type: none"> <li>○ Using SQL to Work with Database Data, Creating Your Own Tables</li> </ul> </li> </ul>	
<b>Unit – 2</b>	<b>Introduction to MVC</b> <ul style="list-style-type: none"> <li>• Introducing ASP.NET MVC 4.0 <ul style="list-style-type: none"> <li>○ What Is ASP.NET?   ASP.NET Web Pages   ASP.NET MVC  </li> </ul> </li> <li>• Installation <ul style="list-style-type: none"> <li>○ Software Requirements for ASP.NET MVC4   Installing ASP.NET MVC 4 Server Components   Visual Studio Application Templates   Anatomy of ASP.NET MVC 4 Internet Application</li> </ul> </li> <li>• ASP.NET MVC 4 Web Application   Description of the Application</li> </ul>	<b>20%</b>
<b>Unit – 3</b>	<b>MVC Architecture &amp; Forms and HTML Helpers</b> <ul style="list-style-type: none"> <li>• Understanding Controllers <ul style="list-style-type: none"> <li>○ The Routing Engine   Creating Controllers   Working with Action Methods</li> </ul> </li> <li>• Understanding Views <ul style="list-style-type: none"> <li>○ View Engines   Working with Views   The Rendering Process   Understanding the Razor View Engine   Working with View Data and View Bag   Working with Strongly Typed Views   Understanding ASP.NET MVC Mobile features</li> </ul> </li> <li>• Understanding Models <ul style="list-style-type: none"> <li>○ What Are Models?   Adding a Business Model   Adding View Models   Understanding Model Binding</li> </ul> </li> <li>• Using Forms and HTML Helpers <ul style="list-style-type: none"> <li>○ The Action and the Method, To GET or to POST?</li> <li>○ HTML Helpers, Other Input Helpers, Rendering Helpers</li> </ul> </li> </ul>	<b>20%</b>

<b>Unit – 4</b>	<b>Data, AJAX, jQuery, &amp; Web API</b> <ul style="list-style-type: none"> <li>Data validation <ul style="list-style-type: none"> <li>The Validation Workflow   Manual Validation   Validation with Data Annotations   Creating Custom Data Annotations</li> </ul> </li> <li>AJAX and JQuery <ul style="list-style-type: none"> <li>Introducing JQuery   Understanding Unobtrusive Javascript   Working with Ajax   Working with JSON   Introducing Web API</li> </ul> </li> <li>Security <ul style="list-style-type: none"> <li>Authentication and Authorization  Securing Controllers and Action Methods   Authenticating with External Sources   Implementing Membership and Roles   Securing ASP.NET MVC Applications Against External Attacks</li> </ul> </li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Testing Application</b> <ul style="list-style-type: none"> <li>Routing <ul style="list-style-type: none"> <li>Routing Concepts   Creating Custom Routs   Creating a Catch-all Segment   Adding Constraints to Routes   Understanding when Routing is Not Applied   ASP.NET Routing Vs. URL Rewriting   Generating Links and URLs</li> </ul> </li> <li>Testing the Application <ul style="list-style-type: none"> <li>Understanding Unit Testing   Examining the Test Project   Testing Controllers   Testing Routes</li> </ul> </li> </ul>	<b>20%</b>

### Learning Outcome:

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

LO1: Developing web application

LO2: Developing web applications using MVC Architecture

LO3: Working with Entity Framework

LO4: Implementing JSON, AJAX, JQUERY and Web API

### Text Book:

1. ASP.NET 4.0 Black Book, Dreamtech Press
2. Beginning ASP.NET MVC 4, Jose Rolando Guay Paz, Apress

**Reference Books & Web Links:**

1. Pro Asp.Net MVC 5, Adam Freeman, Apress
2. Professional ASP.NET MVC 5, Jon Galloway, Brad Wilson, K. Scott Allen, David Matson, Wrox Publication
3. <http://www.asp.net/web-api/overview/older-versions>
4. <http://www.asp.net/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/implementing-basic-crud-functionality-with-the-entity-framework-in-asp-net-mvc-application>

**Chapter wise coverage :**

Unit 1: Chapter No 1, 2, 4, 6, 10 of Text Book-1

Unit 2: Chapter No 1,2 , 3 of Text Book-2

Unit 3: Chapter No 4, 5, 6 of Text Book – 2

Unit 4: Chapter No 7,8, 9 of Text Book – 2

Unit 5: Chapter No 10,11,12 of Text Book-2

**List of Experiments**

1. Create a registration page and store the details in the database
2. Develop an application to create user profile and allow the user to update the profile.  
After getting the input data from the user, the profile data to be displayed in the correct format.

**Getting Started with ASP.NET MVC 4**

1. Intro to ASP.NET MVC 4
2. Adding a Controller
3. Adding a View
4. Adding a Model
5. Accessing Your Model's Data from a Controller
6. Examining the Edit Methods and Edit View
7. Adding a New Field to the Movie Model and Table
8. Adding Validation to the Model

## 9. Examining the Details and Delete Methods

### **Getting Started with EF5 using MVC 4**

1. Creating an Entity Framework Data Model
2. Implementing Basic CRUD Functionality
3. Sorting, Filtering, and Paging
4. Creating a More Complex Data Model
5. Reading Related Data
6. Updating Related Data
7. Handling Concurrency
8. Implementing Inheritance
9. Implementing the Repository and Unit of Work Patterns
10. Advanced Entity Framework Scenarios

### **API**

1. Enabling CRUD Operation in Web API 1
2. Using ASP.NET Web API1 with EF5
3. Self-Host Web API 1(C#)
4. Build RESTful API's with ASP.NET Web API

## MCA230204 – Software Engineering

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
3	2	1	0	60	0	40	0	100

### Prerequisites:

- Knowledge of basic computer science.

### Course Objectives:

The objective of the course is to,

- To introduce fundamentals of software engineering including requirement specifications, software design, testing, and maintenance

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Introduction</b> <ul style="list-style-type: none"> <li>Introduction: Introduction to software Engineering, Software characteristics, Software components, Software applications, Software Engineering Principles, Software metrics and measurement, monitoring and control.</li> <li>Software development life-cycle, Water fall model, prototyping model, Incremental model, Iterative enhancement Model, Spiral model.</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Software Requirement Specification</b> <ul style="list-style-type: none"> <li>Requirements Elicitation Techniques, Requirements analysis, Models for Requirements analysis, requirements specification, and Requirements validation.</li> </ul>	<b>20%</b>
<b>Unit – 3</b>	<b>System Design</b> <ul style="list-style-type: none"> <li>Design Principles: Problem partitioning, abstraction. Top-down and bottom-up – design, structured approach.</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>Functional versus the object-oriented approach of design, design specification, Cohesiveness, and Coupling.</li> </ul>	
<b>Unit – 4</b>	<b>Software Project Management:</b> <ul style="list-style-type: none"> <li>Project planning and Project scheduling.</li> <li>Software Metrics: Size Metrics like LOC, Token Count, Function Count.</li> <li>Cost estimation using models like COCOMO.</li> <li>Risk management activities.</li> </ul> <b>Software Reliability and Quality Assurance</b> <ul style="list-style-type: none"> <li>Reliability issues, Reliability metrics, reliability models, Software quality, ISO 9000 certification for the software industry, SEI capability maturity model.</li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Testing</b> <ul style="list-style-type: none"> <li>Verification and validation, code inspection, test plan, and test case specification.</li> <li>Level of testing: Unit, Integration Testing, Top-down, and bottom-up integration testing, Alpha and Beta testing, System testing, and debugging. functional testing, structural testing, and Software testing strategies.</li> </ul> <b>Software Maintenance</b> <ul style="list-style-type: none"> <li>Structured Vs unstructured maintenance, Maintenance Models, Configuration Management, Reverse Engineering, Software Re-engineering.</li> </ul>	<b>20%</b>

### Learning Outcome:

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

LO1: Various software characteristics and analyze different software Development Models.

LO2: Demonstrate SRS and apply basic software quality assurance practices.

LO3: Compare and contrast various methods for software design.

LO4: Formulate a testing strategy for software systems

LO5: Various software management tools for development, maintenance, and analysis.

**Text Book:**

1. K.K. Aggarwal & Yogesh Singh, “Software Engineering”, New Age International Publishers, New Delhi, 2003.
2. R S Pressman, “Software Engineering: A Practitioners Approach”, McGraw Hill

**Reference Books:**

1. Ian Sommerville, “Software Engineering”, Addison Wesley

**Chapter wise coverage:**

Unit 1: Chapter No 1,2 of Text Book-1, Chapter No 1(Principles of Software Engineering),4(Incremental model) of Text book 2

Unit 2: Chapter No 3 of Text Book-1

Unit 3: Chapter No 5, of Text Book – 1

Unit 4: Chapter No 4, 6, 7 of Text Book – 1

Unit 5: Chapter No 8, 9 of Text Book-1



## MCA230205 – Cyber Security

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
3	2	1	0	60	0	40	0	100

### Prerequisites:

- Fundamental knowledge of Networking

### Course Objectives:

- To learn the concepts behind the security threats impacting
- It provides practical information and suggestions for common security problems
- To gain knowledge of doing independent study and research in the field of cyber security.

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Cyber Security Fundamentals</b> <ul style="list-style-type: none"> <li>Network and Security Concepts</li> <li>Information Assurance Fundamentals</li> <li>Basic Cryptography</li> <li>Symmetric Encryption</li> <li>Public Key Encryption</li> <li>Domain Name System</li> <li>Firewalls</li> </ul>	<b>20%</b>
<b>Unit – 2</b>	<b>Attacker Techniques and Motivations</b> <ul style="list-style-type: none"> <li>How Hacker Cover Their Tracks</li> <li>Tunnelling techniques</li> <li>Fraud techniques</li> <li>Phishing, Smishing, Vishing, and Mobile malicious code</li> <li>Rogue Antivirus</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>Click Fraud</li> <li>Threat Infrastructure</li> </ul>	
<b>Unit – 3</b>	<b>Exploitation</b> <ul style="list-style-type: none"> <li>Techniques to gain a Foothold</li> <li>Shell code</li> <li>Integer Overflow Vulnerabilities</li> <li>Stack Based Buffer Overflows</li> <li>Format String Vulnerabilities</li> <li>SQL Injection</li> <li>Malicious PDF Files</li> <li>Race Condition</li> <li>Web Exploit Tools</li> <li>DOS Conditions</li> <li>Brute Force and Dictionary Attack</li> <li>Misdirection, Reconnaissance, and Disruption Methods</li> </ul>	<b>20%</b>
<b>Unit – 4</b>	<b>Malicious Code</b> <ul style="list-style-type: none"> <li>Self-Replicating Malicious Code</li> <li>Evading Detection and Elevating Privileges</li> <li>Persistent Software Techniques</li> <li>Rootkits</li> <li>Spyware</li> <li>Attack against Privileged User accounts and Escalation</li> <li>Token Kidnapping</li> <li>Virtual Machine Detection</li> <li>Stealing Information and Exploitation</li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Defense and Analysis Techniques</b> <ul style="list-style-type: none"> <li>Memory Forensics</li> <li>Honeypots</li> <li>Malicious Code Naming</li> <li>Automated Malicious Code Analysis Systems</li> <li>Intrusion Detection Systems</li> </ul>	<b>20%</b>

**Learning Outcome:**

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

LO1: To learn the concepts of the security threats impacting

LO2: Explains practical information and suggestions for common security problems

LO3: To gain knowledge of doing independent study and research in the field of cyber security.

**Text Book:**

1. Cyber Security Essentials: CRC Press By James Graham, Richard Howard, Ryan Olson.

**Reference Books:**

1. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives – Nina Godbole, Sunit Belapur, Wiley India Publications Released: April 2011
2. Internet Forensics: Using Digital Evidence to Solve Computer Crime Robert Jones, O'Reilly Media, Released: October 2005
3. Windows Forensics: The field guide for conducting corporate computer investigations Chad Steel, Wiley India Publications Released: December 2006

**Chapter wise coverage :**

Unit 1: Chapter No. 1 of Text Book

Unit 2: Chapter No. 2 of Text Book

Unit 3: Chapter No. 3 of Text Book

Unit 4: Chapter No. 4 of Text Book

Unit 5: Chapter No. 5 of Text Book

## MCA230206 – Software Testing & Quality Assurance

Subject Total Credit	Teaching scheme (per week)			Examination scheme				
				Internal [60%]		External [40%]		Total
	Theory	Tutorial	Practical	Theory	Practical	Theory	Practical	
3	2	1	0	60	0	40	0	100

### Prerequisites:

- Software Engineering Basic

### Course Objectives:

- To understand software testing & quality assurance approaches.
- To get the knowledge of the various techniques of White box and Black box Testing
- Understanding level s & types of Testing
- Designing test cases using manual testing
- To understand manual & automated Software testing

### Course Contents:

Unit	Content	Weightage
<b>Unit – 1</b>	<b>Introduction to Software Testing</b> <ul style="list-style-type: none"> <li>• Introduction</li> <li>• Evolution</li> <li>• Software Testing Myths and Facts</li> <li>• Software Testing Definitions</li> <li>• Models for Software Testing</li> </ul> <b>Software Testing Terminology and Methodology</b> <ul style="list-style-type: none"> <li>• Software Testing Terminology</li> <li>• Software Testing Life Cycle</li> <li>• Software Testing Life Methodology</li> </ul> <b>Verification and Validation</b> <ul style="list-style-type: none"> <li>• V&amp;V Activities</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>• Verification</li> <li>• Verification of requirements</li> <li>• Verification of High-level Design</li> <li>• Verification of Low-level Design</li> <li>• Validation</li> </ul>	
<b>Unit – 2</b>	<p><b>Dynamic Testing: Black Box Testing</b></p> <ul style="list-style-type: none"> <li>• Boundary Value Analysis</li> <li>• Equivalence Class Testing</li> <li>• State table Base Testing</li> <li>• Decision table Base Testing</li> </ul> <p><b>Dynamic Testing: White Box Testing</b></p> <ul style="list-style-type: none"> <li>• Need of White Box Testing</li> <li>• Logic Coverage Criteria</li> <li>• Basis Path Testing</li> <li>• Graph Matrices</li> <li>• Loop Testing</li> <li>• Data Flow Testing</li> <li>• Mutation Testing</li> </ul>	<b>20%</b>
<b>Unit – 3</b>	<p><b>Static Testing</b></p> <ul style="list-style-type: none"> <li>• Inspection</li> <li>• Structured Walkthrough</li> <li>• Technical Reviews</li> </ul> <p><b>Validation Activities</b></p> <ul style="list-style-type: none"> <li>• Unit Validation Testing</li> <li>• Integration Testing</li> <li>• Function Testing</li> <li>• System Testing</li> <li>• Acceptance Testing</li> </ul> <p><b>Regression Testing</b></p> <ul style="list-style-type: none"> <li>• Progressive vs Regressive Testing</li> <li>• Objectives of Regressive Testing</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"> <li>Types of Regressive Testing</li> <li>Regression testing Techniques</li> </ul>	
<b>Unit – 4</b>	<b>Test Management</b> <ul style="list-style-type: none"> <li>Test Organization</li> <li>Structure of Testing Group</li> <li>Testing Planning</li> <li>Detailed Test Design and Test Specification</li> </ul> <b>Testing Metrics for Monitoring and Controlling</b> <ul style="list-style-type: none"> <li>Measurement Objectives for Testing</li> <li>Attributes and Corresponding Metrics in Software Testing</li> <li>Attributes</li> <li>Estimation Models for Estimating Testing Efforts</li> <li>Architectural Design Metric Used for Testing</li> <li>Information Flow Metrics Used for Testing</li> <li>Cyclomatic Complexity Measures for Testing</li> <li>Function Point Metrics for Testing</li> <li>Test Point Analysis (TPA)</li> <li>Some Testing Metrics</li> </ul>	<b>20%</b>
<b>Unit – 5</b>	<b>Software Quality Management</b> <ul style="list-style-type: none"> <li>Software Quality</li> <li>Broadening the Concept of Quality</li> <li>Quality Cost</li> <li>Benefits of Investment on Quality</li> <li>Quality Control and Quality Assurance</li> <li>Quality Management (QM)</li> <li>QM and Project Management</li> <li>Quality Factors</li> <li>Methods of Quality Management</li> <li>Software Quality Metrics</li> <li>SQA Models</li> </ul> <b>Automation and Testing Tools</b> <ul style="list-style-type: none"> <li>Need of Automation</li> </ul>	<b>20%</b>

	<ul style="list-style-type: none"><li>• Categorization of Testing Tools</li><li>• Selection of Testing Tools</li><li>• Costs Incurred in Testing Tools</li><li>• Guidelines For Automated Testing</li></ul>	
--	---	--

**Learning Outcome:**

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

LO1: Shows practical approach of designing the test cases and to assess the risk

LO2: Shows the testing models

LO3: Verification and validation as the major components of software testing

LO4: Software testing life cycle along with bug classification and bug life cycle

LO5: Categorization of software testing techniques such as static testing and dynamic testing

LO6: Extensive coverage of regression testing, software testing metrics, and test management

LO7: Efficient test suite to prioritize test cases suitable for a project and appropriate use of testing tools

LO8: Software quality management and test maturity model (TMM)

**Text Book:**

1. Software Testing: Principles and Practices, Oxford By Naresh Chauhan

**Reference Books:**

2. Software Testing: Principles and Practice by Srinivasan Desikan, Gopalaswamy Ramesh, Pearson

**Chapter wise coverage :**

Unit 1: Chapters No. 1, 2, and 3 of Text Book

Unit 2: Chapters No. 4 and 5 of Text Book

Unit 3: Chapter No. 6, 7 and 8 of Text Book

Unit 4: Chapter No. 9 and 11 of Text Book

Unit 5: Chapter No. 13 and 15 of Text Book