CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

ACADEMIC REGULATIONS
&
SYLLABUS 2024 -25

Faculty of Computer Science & Applications

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications

Online Master of Computer Applications
(ONLINE M.C.A.) Programme
(as per CHOICE BASED CREDIT SYSTEM- (CBCS))

CMPICA
Smt. Chandaben Mohanbhai Patel Institute of Computer Applications

Charotar University of Science and Technology (CHARUSAT)
CHARUSAT Campus, At Post: Changa – 388421, Taluka: Petlad, District: Anand
Phone: 02697-247500, Fax: 02697-247100, Email: info@charusat.ac.in
www.charusat.ac.in

ACADEMIC RULES

To ensure uniform system of education, duration of post graduate programmes, eligibility criteria for and mode of admission, credit load requirement and its distribution between course and system of examination and other related aspects, following academic rules and regulations are recommended.

1. **System of Education**

   The Semester system of education should be followed across the Charotar University of Science and Technology (CHARUSAT) at Master's levels. Each semester will be at least 90 working days' duration. Every enrolled student will be required to take a specified load of course work in the chosen course of specialization and also complete a project/dissertation if any.

2. **Duration of Programme**

Postgraduate Programme	Master of Computer Applications(MCA-ONLINE)
Minimum	4 Semesters (2 Academic years)
Maximum	6 Semesters (3 Academic years)

3. **Eligibility & Mode of Admissions**

   Eligibility of a candidate and mode of admission to the programme will be according to the regulations for admission committee.

4. **Programme structure and Credits**

   A student admitted to a program should study the course and earn credits specified in the course structure.

5. Course Evaluation

5.1 The performance of every student in each course will be evaluated as follows:

5.1.1 Internal evaluation by the course faculty member(s) based on continuous assessment, for 30% of the marks for the course.

5.1.2 Final examination by the University through online mode or practical test or oral test or presentation by the student or a combination of any two or more of these, for 70% of the marks for the course.

5.2 University Examination

5.2.1 The final examination by the University for 70% of the evaluation for the course will be through online mode or practical test or oral test or presentation by the student or a combination of any two or more of these.

5.2.2 In order to earn the credit in a course a student has to obtain grade other than FF.

5.2.3 Performance at Internal & University Examination will be done on the relative grading system.

6. Grading

The student's performance in any semester will be assessed by the Semester Grade Point Average (SGPA). Similarly, his performance at the end of two or more consecutive semesters will be denoted by the Cumulative Grade Point Average (CGPA). The SGPA and CGPA are defined as follows:

Grading Scheme Range of Marks (%)	96.0-100	86.0-95.9	76.0-85.9	66.0-75.9	56.0-65.9	46.0 – 55.9	36.0 – 45.9	Below 36.0	Absent
Letter Grade	**O** (Outstanding)	**A+** (Excellent)	**A** (Very Good)	**B+** (Good)	**B** (Above Average	**C** (Average)	**P** (Pass)	**F** (Fail)	**Ab** (Absent)
Grade Point	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**0**	**0**

SGPA = ΣCiGi/ΣCi where Ci is the number of credits of course i Gi is the Grade Point for the course i and i = 1 to n, n = number of courses in the semester.

CGPA =Σ CiGi/ΣCi where Ci is the number of credits of course i Gi is the Grade Point for the course i and i = 1 to n, n = number of courses of all semesters up to which CGPA is computed.

The minimum passing marks for each pattern of evaluation are 36% .

7. Detention Rule

A student will be promoted to next year only if he/she has cleared all the courses of the year he/she is studying in.

Awards of Degree: Every student of the programme who fulfils the following criteria will be eligible for the award of the degree:

7.1.1 He should have earned at least minimum required credits as prescribed in course structure.

7.1.2 He should have cleared all evaluation components in every course.

8. Award of Class

The class awarded to a student in the programme is decided by the final CGPA as per the following scheme:

Distinction:	**CGPA ≥7.0 & ≤10**
First class:	**CGPA ≥ 6.0 & <7.0**
Second Class:	**CGPA ≥ 5.0 & <6.0**
Pass Class:	**CGPA <5.0**

9. Transcript

The transcript issued to the student at the time of leaving the University will contain a consolidated record of all the courses taken, credits earned, grades obtained, SGPA, CGPA, class obtained, etc.

CHOICE BASED CREDIT SYSTEM

The choice based credit system provides flexibility in designing curriculum and assigning credits based on the course content and hour of teaching. The choice based credit system provides an opportunity for the students to choose courses from the prescribed courses comprising core, elective and open elective courses. The CBCS provides a cafeteria type approach in which the students can take courses of their choice and adopt an interdisciplinary approach to learning. The courses shall be evaluated on the grading system, which is considered to be better than the conventional marks system.

CBCS – Conceptual Definitions / Key Terms (Terminologies)

Types of Courses: The Programme Structure consist of 3 types of courses:

Foundation Courses, Core Courses and Elective Courses.

Foundation Course

These courses are offered by the institute in order to prepare students for studying courses to be offered at higher levels.

Core Courses

A Course which shall compulsorily be studied by a candidate to complete the requirements of a degree / diploma in a said programme of study is defined as a core course. Following core courses are incorporated in CBCS structure:

A. University Core Courses(UC):

University core courses are compulsory courses which are offered across university and must be completed in order to meet the requirements of programme.

B. Programme Core Courses(PC):

Programme core courses are compulsory courses offered by respective programme owners, which must be completed in order to meet the requirements of programme.

Elective Courses

Generally, a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline of study or which provides an extended scope or which enables an exposure to some other discipline / domain or nurtures the candidate's proficiency / skill is called an elective course. Following elective courses are incorporated in CBCS structure:

A. University Elective Courses(UE):

The pool of elective courses offered across all faculties / programmes.

B. Institute Elective Course (IE)

Institute elective courses are those courses which any students of the University/Institute of a Particular Level (PG/UG) will choose as offered or decided by the University/Institute from time-to-time irrespective of their Programme /Specialization.

C. Programme Elective Courses(PE):

The programme specific pool of elective courses offered by respective programme.

Vision, Mission, PEOs, POs and PSOs

Vision

To become a leading institution in the field of computer applications and contribute in national efforts of computerizing public systems.

Mission

To produce competent computer professionals with the ability to face future challenges.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

The post graduates will:

PEO1: Be able to understand the requirement of computing problem and implement an effective solution.

PEO2: Be able to successfully take up various available career options.

PEO3: Be able to continuously learn in their preferred domains.

PEO4: Be able to acquit themselves in ethical and professional manner while demonstrating effective skills.

PROGRAM OUTCOMES (POs)

The Post Graduate of Computer Science and Applications will be able to:

PO1. Computational Knowledge: Understand and apply mathematical foundation, computing and domain knowledge for the conceptualization of computing models from defined problems.

PO2. Problem Analysis: Ability to identify, critically analyze and formulate complex computing problems using fundamentals of computer science and application domains.

PO3. Design / Development of Solutions: Ability to transform complex business scenarios and contemporary issues into problems, investigate, understand and propose integrated solutions using emerging technologies

PO4. Conduct Investigations of Complex Computing Problems: Ability to devise and conduct experiments, interpret data and provide well informed conclusions.
PO5. Modern Tool Usage: Ability to select modern computing tools, skills and techniques necessary for innovative software solutions

PO6. Professional Ethics: Ability to apply and commit professional ethics and cyber regulations in a global economic environment.

PO7. Life-long Learning: Recognize the need for and develop the ability to engage in continuous learning as a Computing professional.

PO8. Project Management and Finance: Ability to understand, management and computing principles with computing knowledge to manage projects in multidisciplinary environments.

PO9. Communication Efficacy: Communicate effectively with the computing community as well as society by being able to comprehend effective documentations and presentations.

PO10. Societal & Environmental Concern: Ability to recognize economical, environmental, social, health, legal, ethical issues involved in the use of computer technology and other consequential responsibilities relevant to professional practice.

PO11. Individual & Team Work: Ability to work as a member or leader in diverse teams in multidisciplinary environment.

PO12. Innovation and Entrepreneurship: Identify opportunities, entrepreneurship vision and use of innovative ideas to create value and wealth for the betterment of the individual and society.

PROGRAM SPECIFIC OUTCOMES (PSOs)

At the end of the programme, the student should be able to

PSO1: Analyze, design and develop effective software applications for solving contemporary challenges.

PSO2: Harness the necessary skills with spirit of research and entrepreneurship.

Total Credits of Programme (4 Semesters): 108

TEACHING SCHEME
&
EXAMINATION SCHEME

FOR

ONLINE MCA PROGRAMME

EFFECTIVE FROM
ACADEMIC YEAR 2024-25

Teaching & Examination Scheme
Online Master of Computer Applications (M.C.A.) Programme
(Choice Based Credit System)

Effective from Year 2024 - 25
Semester – I

Course Code	Course Title	Teaching Scheme				Examination Scheme				
		Contact Hours			Credit	Theory		Practical		Total
		Theory	Practical	Total		CCE	SEE	CCE	SEE	
CAOE501 CAME503 CAME504	Elective-I	3	-	3	3	30	70	-	-	100
CAOC505	Web Development using Open Source Technologies	2	3	5	5	30	70	30	70	200
CAOC506	Enterprise Computing using Java EE	3	3	6	6	30	70	30	70	200
CAOC507	Programming with .NET Architecture	2	3	5	5	30	70	30	70	200
CAOC508	Database Technologies	3	3	6	6	30	70	30	70	200
HSUS501	Academic Speaking and Presentation Skills	-	3	3	3	-	-	30	70	100
		13	15	28	28	500		500		1000

*CCE- Continuous and Comprehensive Evaluation (Assignments, Unit Tests, Sessional Test, Case Study, Attendance, Practical Test, Viva etc.) *SEE- Semester –End- Evaluation

Elective-I	
Course Code	**Course Title**
CAOE501	Cloud Computing
CAME503	Ethical Hacking
CAME504	Cyber Security and Privacy

Teaching & Examination Scheme
Online Master of Computer Applications (M.C.A.) Programme
(Choice Based Credit System)

Effective from Year 2024 - 25
Semester-II

Course Code	Course Title	Teaching Scheme				Examination Scheme				
		Contact Hours			Credit	Theory		Practical		Total
		Theory	Practical	Total		CCE	SEE	CCE	SEE	
CAOE510	Python Web framework	3	3	6	6	30	70	30	70	200
CAOC513	Advanced Mobile Programming	3	3	6	6	30	70	30	70	200
CAOC514	Full-Stack Web Development	3	3	6	6	30	70	30	70	200
CAOC515	Software Engineering with Agile and DevOps	3	-	3	3	30	70	-	-	100
HSUA501	Academic Writing	-	3	3	3	-	-	30	70	100
		12	12	24	24	400		400		800

*CCE- Continuous and Comprehensive Evaluation (Assignments, Unit Tests, Sessional Test, Case Study, Attendance, Practical Test, Viva etc.) *SEE- Semester –End- Evaluation

Elective-II	
Course Code	**Course Title**
CAOE510	Python Web framework

Teaching & Examination Scheme
Online Master of Computer Applications (M.C.A.) Programme
(Choice Based Credit System)

Effective from Year 2024 - 25
Semester-III

Course Code	Course Title	Teaching Scheme				Examination Scheme				
		Contact Hours			Credit	Theory		Practical		Total
		Theory	Practical	Total		CCE	SEE	CCE	SEE	
CAOE601 – CAOE604	Elective-III	3	3	6	6	30	70	30	70	200
CAOC605	Software Quality Assurance	3	3	6	6	30	70	30	70	200
CAOC606	Mini Project	-	12	12	12	-	-	120	280	400
CAOC607	Green Computing		2	2	2	-	-	30	70	100
		06	20	26	26	200		700		900
*CCE- Continuous and Comprehensive Evaluation (Assignments, Unit Tests, Sessional Test, Case Study, Attendance, Practical Test, Viva etc.) *SEE- Semester –End- Evaluation										

Elective-III	
Course Code	**Course Title**
CAOE601	Digital Image Processing
CAOE602	Object Oriented Programming for Smart Contracts
CAOE603	Data Analytics
CAOE604	Internet of Things

Teaching & Examination Scheme
Online Master of Computer Applications (M.C.A.) Programme
(Choice Based Credit System)

Effective from Year 2024 - 25
Semester-IV

Course Code	Course Title	Teaching Scheme				CCE	SEE		Total
		Contact Hours			Credit	Continuous Evaluation	Report	Presentation & Viva	
		Inst.	Industry	Total					
CAOC608	Dissertation / Project Work	2	28	30	30	240	280	280	800
*CCE- Continuous and Comprehensive Evaluation (Assignments, Unit Tests, Sessional Test, Case Study, Attendance, Practical Test, Viva etc.) *SEE- Semester –End- Evaluation									

DETAILED SYLLABUS

Faculty of Computer Science & Applications

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications

ONLINE MCA PROGRAMME
EFFECTIVE FROM
ACADEMIC YEAR 2024-25

TEACHING SCHEME & DETAILED SYLLABUS

FOR

ONLINE MCA PROGRAMME

(Semester-1)

EFFECTIVE FROM ACADEMIC YEAR 2024-25

Teaching & Examination Scheme
Online Master of Computer Applications (M.C.A.) Programme
(Choice Based Credit System)

Effective from Year 2024 - 25
Semester – I

Course Code	Course Title	Teaching Scheme				Examination Scheme				
		Contact Hours			Credit	Theory		Practical		Total
		Theory	Practical	Total		CCE	SEE	CCE	SEE	
CAOE501 CAME503 CAME504	Elective-I	3	-	3	3	30	70	-	-	100
CAOC505	Web Development using Open Source Technologies	2	3	5	5	30	70	30	70	200
CAOC506	Enterprise Computing using Java EE	3	3	6	6	30	70	30	70	200
CAOC507	Programming with .NET Architecture	2	3	5	5	30	70	30	70	200
CAOC508	Database Technologies	3	3	6	6	30	70	30	70	200
HSUS501	Academic Speaking and Presentation Skills	-	3	3	3	-	-	30	70	100
		13	15	28	28	500		500		1000

*CCE- Continuous and Comprehensive Evaluation (Assignments, Unit Tests, Sessional Test, Case Study, Attendance, Practical Test, Viva etc.) *SEE- Semester –End- Evaluation

Elective-I	
Course Code	**Course Title**
CAOE501	Cloud Computing
CAME503	Ethical Hacking
CAME504	Cyber Security and Privacy

CAOE501 - Cloud Computing

Pre-requisite: Operating System Concepts and Network Technology

Methodology & Pedagogy: During theory lectures the emphasis will be given on the fundamentals of cloud computing. Students will be introduced basic types, architecture, service providers, mechanism, security issues and some hidden aspects of cloud computing. Students will give practical exposure in form of case study and by showing cloud infrastructure of university.

Outline of the course:

Unit Number	Title of the Unit	Minimum Number of Theory Hours
1	Evolution of Cloud Computing	04
2	Understanding Cloud Computing and basic types	07
3	Fundamentals of Cloud Architecture and Service Providers	07
4	Cloud Computing Mechanisms	07
5	Cloud Computing Security and Business Use	06
6	Intensive analysis of Cloud Computing with case studies	05

Total Hours (Theory): 36
Total Hours:36

Detail Syllabus:

Unit I : Evolution of Cloud Computing **Hours 04**

Introduction of Cloud Computing, Growth of technology, Paradigm Shift in Computing, distributed nature of service Provisioning, Support entrepreneurship using Cloud Computing.

Unit II : Understanding Cloud Computing and basic types **Hours 07**

Advantages and drawbacks of Cloud Computing, Essential component for Cloud contract, Major outage of Cloud Computing and Enhancers for Cloud Computing. Introduction to SaaS, PaaS, IaaS. Introduction to Public Cloud, Private Cloud, Hybrid Cloud and Community Cloud, Storage Services for Cloud Computing.

Unit III : Fundamentals of Cloud Architecture and Service Providers **Hours 07**

Workload Distribution Architecture, Resource Pooling Architecture, Dynamic Scalability Architecture, Elastic Resource Capacity Architecture, Service Load Balancing Architecture, Cloud Bursting Architecture, Elastic Disk Provisioning Architecture, Redundant Storage Architecture. Introduction to major Cloud Service Provider: Amazon Web Services, Google Cloud. Microsoft Windows Azure and Office 365, Hp Cloud, RackSpace, CSC Corp, Verizon Terrimark, DropBox.

Unit IV : Cloud Computing Mechanisms **Hours 07**

Introduction to Cloud Infrastructure Mechanisms: Logical Network Perimeter, Virtual Server, Cloud Storage Device, Cloud Storage Levels, Network Storage Interfaces, Object Storage Interfaces, Database Storage Interfaces, Relational Data Storage, on-Relational Data Storage, Cloud Usage Monitor, Monitoring Agent, Resource Agent, Polling Agent, Resource Replication. Introduction to Cloud Management Mechanisms: Remote Administration System, Resource Management System, SLA Management System, Billing Management System.

Unit V : Cloud Computing Security and Business Use **Hours 06**

Introduction to Encryption, Symmetric Encryption, Asymmetric Encryption, Hashing, Digital Signature, Public Key Infrastructure (PKI), Identity and Access Management (IAM), Single SignOn (SSO), Cloud-Based Security Groups. Overview of Compliance and Certification, Access Control, Organizational Control. Benefits of Business using Cloud Computing, Risk of Cloud Computing, Cost factor in Cloud Computing.

Unit VI : Intensive analysis of Cloud Computing with case studies **Hours 05**

Overview of Cloud services, Designing Solutions for the Cloud, Implement & Integrate Solutions, Emerging Markets and the Cloud

Core Books:

1. Kevin L. Jackson, Scott Goessling: Architecting Cloud Computing Solutions: Packt Publication: 2018

2. Thomas Erl, Zaigham Mahmood and Ricardo Puttini: Cloud Computing Concepts, Technology & Architecture, PHI, 2013.
3. S. Srinivasan: Cloud Computing Basics, Springer,2014.

Reference Books:

1. Derrick Rountree, Ileana Castrillo : The Basics of Cloud Computing, Syngress, 2013.
2. Rajkumar Buyya, James Broberg, Andrzej M. Goscinsk: Cloud Computing- Principles and Paradigms, John Wiley &Sons, 2011.

Web References:

1. http://whatisCloud.com/basic_concepts_and_terminology/Cloud [For basic terminology of Cloud Computing]
2. http://www.tutorialspoint.com/Cloud_Computing/ [For cloud computing lecture notes]
3. http://www.intel.in/content/dam/www/public/us/en/documents/guides/cloudcomputingvirtualization-building-private-iaas-guide.pdf [For cloud computing virtualization]
4. www.cs.purdue.edu/.../Anya-Kim-Bhargava-MCCWorkshop.ppt [Security issues PPTs]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Students will learn basics of cloud computing
CO2 :	The students will be familiar with various cloud architectures and services.
CO3 :	They will get the knowledge of various network mechanics.
CO4 :	Students will get various business aspects of cloud computing with security aspects
CO5 :	They will be able to design and deploy Cloud Infrastructure

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Evolution of Cloud Computing	√				
2	Understanding Cloud Computing and basic types		√			
3	Fundamentals of Cloud Architecture and Service Providers		√			
4	Cloud Computing Mechanisms			√		
5	Cloud Computing Security and Business Use				√	
6	Hidden Aspects of Cloud Computing					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	-	-	1	-	3	1	2	2	-	-	1	-
CO2	3	1	1	1	1	1	3	1	2	2	1	3	2	-
CO3	3	2	1	2	1	3	3	2	2	2	-	3	2	-
CO4	3	3	2	3	1	3	3	2	2	2	3	-	1	-
CO5	3	3	3	3	1	3	3	3	2	2	2	-	3	1

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAME503 Ethical Hacking

Description:

Credit and Week:

Teaching Scheme	Week	Marks	Credit
	12	100	3

About the course:

Ethical hacking is a subject that has become very important in present-day context, and can help individuals and organizations to adopt safe practices and usage of their IT infrastructure. Starting from the basic topics like networking, network security and cryptography, the course will cover various attacks and vulnerabilities and ways to secure them. There will be hands-on demonstrations that will be helpful to the participants. The participants are encouraged to try and replicate the demonstration experiments that will be discussed as part of the course.

Pre-requisites:

Basic concepts in programming and networking

Industry support:

TCS, Wipro, CTS, Google, Microsoft, Qualcomm

Course layout:

Week 1: Introduction to ethical hacking. Fundamentals of computer networking. TCP/IP protocol stack.

Week 2: IP addressing and routing. TCP and UDP. IP subnets.

Week 3: Routing protocols. IP version 6.

Week 4: Installation of attacker and victim system. Information gathering using advanced google search, archive.org, net craft, whois, host, dig, dnsenum and NMAP tool**.**

Week 5: Vulnerability scanning using NMAP and Nessus. Creating a secure hacking environment. System Hacking: password cracking, privilege escalation, application execution. Malware and Virus. ARP spoofing and MAC attack.

Week 6: Introduction to cryptography, private-key encryption, public-key encryption.

Week 7: Cryptographic hash functions, digital signature and certificate, applications.

Week 8: Steganography, biometric authentication, network-based attacks, DNS and Email security.

Week 9: Packet sniffing using Wireshark and burp suite, password attack using burp suite. Social engineering attacks and Denial of service attacks.

Week 10: Elements of hardware security: side-channel attacks, physical inclinable functions, hardware Trojans.

Week 11: Different types of attacks using Metasploit framework: password cracking, privilege escalation, remote code execution, etc. Attack on web servers: password attack, SQL injection, cross site scripting.

Week 12: Case studies: various attacks scenarios and their remedies.

Books and references

Text Books / Basic Material

1. Data and Computer Communications – W. Stallings.
2. Data Communication and Networking – B. A. Forouzan
3. TCP/IP Protocol Suite – B. A. Forouzan
4. UNIX Network Programming – W. R. Stallings
5. Introduction to Computer Networks and Cybersecurity – C-H. Wu and J. D. Irwin
6. Cryptography and Network Security: Principles and Practice – W. Stallings

Criteria to get a certificate:

Average assignment score = 25% of average of best 8 assignments out of the total 12 assignments given in the course.
Exam score = 75% of the proctored certification exam score out of 100

Final score = Average assignment score + Exam score

YOU WILL BE ELIGIBLE FOR A CERTIFICATE ONLY IF AVERAGE ASSIGNMENT SCORE >=10/25 AND EXAM SCORE >= 30/75. If one of the 2 criteria is not met, you will not get the certificate even if the Final score >= 40/100.

CAME504 Cyber Security and Privacy

Description:

Credit and Week:

Teaching Scheme	Week	Marks	Credit
	12	100	3

About the course:

This course introduces to management students the concepts, technologies, practices and challenges associated with cybersecurity as applied in organizations. Protection as well as disclosure of information pose unique challenges and also allude to economic and technological implications. The course takes a broad view of cyber security along with information privacy by analyzing relevant organizational, human, legal and policy issues. Through this course, students will explore cyber security along with information privacy with a managerial focus. Students are expected to develop a wholesome understanding about cyber security and privacy risks to businesses covering governance, compliance and risk mitigation and closely study certain business domains.

Pre-requisites:

A core course on Management Information Systems desirable (not mandatory).

Industry support:

IT services industry, AI/Block chain start ups, Industry 4.0, autonomous vehicles industry.

Week 1 : Foundations, cyber security, information security and related concepts, Principles of information security management, Confidentiality, Integrity, Availability and related concepts.

Week 2 : Security management, Governance, Risk and Compliance (GRC) , Contingency planning, incidence response, disaster recovery and business continuity.

Week 3 : Understanding security policy, security behavior , Risk management: Risk identification, threat modeling, strategies.

Week 4 : Control strategies and protection mechanisms (Guest lecture), Cryptography for security.

Week 5 : Information security and privacy, Regulatory landscape: Fair information practices, US regulatory frameworks .

Week 6 : Regulatory landscape: EU's GDPR and its implications and other privacy and cyber security regulations, Cyber security and privacy in the Indian context, evolution and issues.

Week 7 : Economics of privacy, privacy calculus and trade-offs, privacy paradox, Managing stakeholders, making choices on security and privacy Course layout:

Books and references

- Michael E. Whitman, Herbert J. Mattord, (2018). Principles of Information Security, 6th edition, Cenage Learning, N. Delhi.
- Darktrace, "Technology" https://www.darktrace.com/en/technology/#machine-learning, accessed November 2018.
- Van Kessel, P. Is cyber security about more than protection? EY Global Information Security Survey 2018-2019.
- Johnston, A.C. and Warkentin, M. Fear appeals and information security behaviors: An empirical study. MIS Quarterly, 2010.
- Arce I. et al. Avoiding the top 10 software security design flaws. IEEE Computer Society Center for Secure Design (CSD), 2014.
- Smith, H. J., Dinev, T., & Xu, H. Information privacy research: an interdisciplinary review. MIS Quarterly, 2011.
- Subramanian R. Security, privacy and politics in India: a historical review. Journal of Information Systems Security (JISSec), 2010.
- Acquisti, A., John, L. K., & Loewenstein, G. **What** is privacy worth? The Journal of Legal Studies, 2013
- Xu H., Luo X.R., Carroll J.M., Rosson M.B. The personalization privacy paradox: An exploratory study of decision making process for location-aware marketing. Decision Support Systems, 2011.

Criteria to get a certificate:

Average assignment score = 25% of average of best 8 assignments out of the total 12 assignments given in the course.

Exam score = 75% of the proctored certification exam score out of 100

Final score = Average assignment score + Exam score

YOU WILL BE ELIGIBLE FOR A CERTIFICATE ONLY IF AVERAGE ASSIGNMENT SCORE >=10/25 AND EXAM SCORE >= 30/75. If one of the 2 criteria is not met, you will not get the certificate even if the Final score >= 40/100.

CAOC505 - Web Development using Open Source Technology

Pre-requisite: Basic understanding of HTML and MySQL.

Methodology & Pedagogy: During theory and practical sessions, students able to install & configure PHP and prerequisite software. Also, student will be emphasized to develop dynamic web applications and Create web API in PHP.

Outline of the course:

Unit No.	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Basics of HTML and JavaScript.	04	36
2	Basics of PHP.	03	
3	Handling Form Data and Database Connectivity with MYSQL	04	
4	PHP Utilities.	05	
5	PHP API client.	03	
6	AJAX With DATABASE.	05	

Total Hours (Theory): 24

Total Hours (Lab): 36

Total Hours: 60

Detail Syllabus:

Unit I: Basics of HTML and Java Script **Hours: 03**

Basics of HTML, Introduction of JavaScript, Variables, Overview of operators, Control statements and looping statement, Overview of DOM, function declaration and calling with event**.**

Unit II: Basics of PHP **Hours: 05**

Introduction to PHP, PHP Apache Web server, Understanding of PHP.INI file Understanding of PHP .htaccess file, PHP Variable, Static & global variable, PHP Operator, Conditional Structure & Looping Structure, Array, User Defined Functions: argument function default argument, variable function, return function.

Unit III: Handling Form Data and Database Connectivity with MYSQL Hours: 05

HTML Form element & its attributes, Send Form data using GET Method & POST Method, Receive Form data using $_GET, $_POST & $_REQUEST variables, PHP form validations, OOPs Concepts, PHP Data Objects (PDO), CRUD operations, Handling Errors, State Management Techniques: Concept of Session, starting session, modifying session variables, Un registering and deleting session variable, Concept of Cookies.

Unit IV: PHP utilities **Hours: 04**

File Uploading: Upload Single and Multiple file using PHP script, Understanding HTTP requests, Exploring and modifying HTTP responses, getting information from web server, Sending mails

Unit V: PHP API client - Hours: 04

Basic API call from PHP, HTTP basics: requests, responses and using cURL, Response codes: get the HTTP status code.

Unit VI: AJAX With DATABASE **Hours: 03**

Basics of AJAX, PHP with AJAX, MySql with AJAX, What is JQuery AJAX, JQuery AJAX with PHP.

Core Books:

1. Vikram Vaswani, PHP: A Beginner's Guide: Indian Edition, First Edition, McGraw Hill, 2009
2. Matt Doyle , Beginning PHP 5.3 , Wrox , 2010
3. Ballard and Moncur, Teach Yourself Javascript in 24 Hours, Sams Publishing, 2015

Reference Books:

1. Timothy Boronczyk , Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael
   K. Glass): Beginning PHP6, Apache, MySQL Web Development: Wrox , 2009.
2. Lynn Beighley and Michael Morrison, Head First PHP & MySQL, First Edition, O'Reilly Publication,
   2009.

Web References:

1. https://github.com/PHPMailer/PHPMailer [PHP Mailer Code]
2. https://www.php.net/ [Official website of PHP]
3. https://www.geeksforgeeks.org/php-tutorials/ [Lecture notes of PHP]
4. https://www.w3schools.com/php/default.asp [Lecture notes of PHP]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Gain understanding of HTML and Java Script.
CO2 :	To utilize knowledge and skills for basics of PHP and functions of PHP
CO3 :	Acquire the knowledge of array and OOP in PHP
CO4 :	Learn Form handling and utilities in PHP.
CO5 :	Be able to develop dynamic web based application using PHP and MySQL with Ajax Call.

Course Outcomes Mapping: -

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Basics of HTML and Java Script	√				
2	Introduction to Open Source Software and PHP, Basics of PHP		√			
3	Control Structure, Function, Array and Overview of OOP in PHP			√		
4	Form Handling Using PHP				√	
5	PHP Utilities				√	
6	Relational Database Using PHP					√

Course Articulation Matrix: -

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	3	-	1	-	1	2	-	-	2	-	3
CO2	3	3	3	3	-	1	-	1	1	-	-	3	-	3
CO3	3	3	3	3	-	1	-	1	1	-	-	3	-	3
CO4	3	3	3	3	-	1	-	1	1	-	-	3	-	3
CO5	3	3	3	3	2	1	1	1	2	1	1	3	3	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOC506 - Enterprise Computing using Java EE

Pre-requisite: Fundamental Knowledge of Core Java (Java SE) concepts.

Methodology & Pedagogy: Advanced Java fundamentals like JDBC, Servlet, JSP and Hibernate provides capabilities and features for developing web application which is essential for developer in order to understand entire web application development life cycle. During the theory sessions of the syllabus, fundamentals of the mentioned technologies will be explained in detail with real time examples. Moreover, practical session will provide hands-on exposure about JDBC, Servlet, JSP and Hibernate with variety of practical assignment. Also, MVC architecture using these technologies will be covered during theory as well as practical sessions.

Outline of the course:

Unit Number	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Fundamentals Object Oriented Concepts	06	36
2	JDBC Programming	06	
3	Web Application Components Fundamentals	05	
4	Servlet API	08	
5	Java Server Pages	06	
6	Basics of Hibernate 4.0	05	

Total Hours (Theory): 36

Total Hours (Lab) : 36

Total Hours: 72

Detail Syllabus:

Unit I : Fundamentals Object Oriented Concepts Hours 06

Object Oriented programming concepts, Classes & Objects, Structure of Java Program, Constructors, Inheritance, Abstract Class, Interface, Exception Handling, Packages, Multithreading, Fundamentals of Collections.

Unit II : Introduction to JDBC Programming Hours 06

The JDBC Connectivity Model, Database Programming: Connecting to the Database, creating a SQL Query, Executing SQL Queries, Updating Database Data, JDBC Types, Error Checking and the SQLException Class, the SQLWarning class, Statement Interface, PreparedStatement, Callable Statement, Result Set Interface, Updatable Result sets, Result SetMetaData, Transaction Management.

Unit III : Fundamentals of Web Application Components Hours 05

Basics of web application development and web components, Introduction of MVC design pattern, Fundamentals of: Containers, Packaging Web Applications, Web Application Structure, JAR Files, WAR Files, HTTP, HTTP request methods, HTML Form Processing, HTTP request response cycle.

Unit IV : Servlet API Hours 08

Servlet Model: Overview of Servlet, Servlet Life Cycle, HTTP Methods Structure and Deployment descriptor ServletContext and ServletConfig interface, Attributes in Servlet.

Request Dispatcher interface, The Filter API: Filter, FilterChain and Filter Config, Cookies and Session Management: Understanding state and session.

Understanding Session Tracking Techniques, Hidden Form Fields, URL Rewriting, Session & Cookies.

Unit V : Java Server Pages Hours 06

JSP Overview: The Problem with Servlets, Life Cycle of JSP Page, JSP Processing, JSP Application Design with MVC, Setting Up the JSP Environment, JSP Directives, JSP Action.

JSP Implicit Objects JSP Form Processing, JSP Session and Cookies Handling, JSP Session Tracking, JSP Database Access, JSP Standard Tag Libraries, JSP Custom Tag, JSP Expression Language, JSP Exception Handling, JSP XML Processing.

Unit VI : Basics of Hibernate 4.0 Hours 05

Overview of Hibernate, Hibernate Architecture, Hibernate Mapping Types, Hibernate O/R Mapping, Hibernate Annotation, Hibernate Query Language.

Core Books:

1. Cay S Horstmann, Gary Cornell: Core Java, Volume II – Advanced Features, 8th Edition, Pearson Education.
2. Marty Hall, Larry Brown: Core Servlets and JavaServer Pages, Volume 2, Advanced Technologies, 2nd Edition, Pearson Education, 2008.
3. Jeff Linwood and Dave Minter: Hibernate 2nd edition Beginning Après publication

Reference Books:

1. Bryan Basham, Kathy Sierra and Bert Bates: Head First Servlet and JSP, O'Reilly Publication, 1st Edition
2. James Keogh: Complete Reference J2EE, Mcgraw publication

Web References:

1. http://courses.coreservlets.com/Course-Materials/csajsp2.html [Servlet Basics]
2. http://www.ceit.es/asignaturas/InteInfo/Recursos/Servlets/JavaServlets.pdf [Servlet Tutorial PDF]
3. http://www.msuniv.ac.in/AdvancedJavaProgrammingwithDatabaseApplication.pdf [JDBC Tutorial]
4. www.doc.ic.ac.uk/~rcheung/teaching/2720/ppt/lecture12.ppt [JSP Tutorial Slides]

Course Outcomes: Upon successful completion of the course, students will:

CO1 :	To make understand the fundamental Concepts of Java
CO2 :	Able to connect Java application with various RDBMS using JDBC
CO3 :	To make understand web application architecture and terminologies
CO4 :	Able to develop real time web application using Servlet and JSP technologies
CO5 :	Able to use Object Relation Mapping using Hibernate to build database.

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Fundamental Concepts of Java	√				
2	JDBC Programming	√	√			
3	Web Application Components Fundamentals	√	√	√		
4	Servlet API			√	√	
5	Java Server Pages			√	√	
6	Basics of Hibernate 4.0					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	2	1	1	-	2	2	-	1	2	2	3	2
CO2	3	2	3	3	-	-	1	-	-	-	2	1	2	2
CO3	3	2	3	2	2	-	-	-	-	2	2	3	3	3
CO4	3	2	3	2	2	2	-	-	1	-	2	1	3	2
CO5	2	2	3	3	2	-	1	1	-	2	3	3	3	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOC507 - Programming with .NET Architecture

Pre-requisite: Familiarity with basic concepts of object-oriented programming

Methodology & Pedagogy:

In the lectures students will learn about basics of .NET framework, ASP.NET and other advanced concepts related to programming with .NET framework for Web Development. In the lab sessions the student will learn about solving real world problems with ASP .NET framework and related technologies to gain experience on Web Development.

Outline of the course:

Unit Number	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Introduction to .NET framework and ASP.NET	02	36
2	ASP .NET Web Forms	05	
3	ADO .NET (Database Connectivity)	08	
4	Introduction to MVC Architecture	02	
5	Application Development using MVC	05	
6	Introduction to Web Services	02	

Total Hours (Theory): 24
Total Hours (Lab):36
Total Hours: 60

Detail Syllabus:

Unit I: Introduction to .NET framework and ASP.NET Hours 02

Introduction to .NET framework, Introduction to Web Programming, Introduction to ASP and ASP.NET.

Unit II: ASP .NET Web Forms Hours 05

Web Forms: "ASPX Page, Code behind File, WebConfig", ASP.NET Page Life Cycle, developing a web form, Working with master pages, State management techniques.

Unit III: ADO .NET (Database Connectivity) Hours 08

Introduction to ADO and ADO.NET, ADO.NET architecture, Connection Oriented and Connection Less architecture, working with Models, Introduction to Three Tier architecture, Working on Data Tier.

Unit IV: Introduction to MVC Architecture Hours 02

Introduction to MVC Architecture, ASP.NET MVC, CRUD operation using ASP .NET MVC architecture, working with web controls in ASP .NET MVC architecture.

Unit V: Application Development using MVC. Hours 05

Execution of ASP .NET MVC Application, Routing in MVC, HTTP Request and Response, Filters, Layout Pages, View Start File.

Unit VI: Introduction to Web Services Hours 02

Introduction to Web Services and RESTful API, Applications of Web API, Understanding AJAX and JSON.

Core Books:

1. Cogent Solutions Inc.: ASP.Net 4.5 Black book, Dreamtech press, 2019
2. Mridila Parihar, Essam Ahmed: ASP .Net Bible, Wiley, 2017
3. Stephon Walther: ASP.Net Unleashed, BPB publication, 2018

Reference Books:

1. Mesbah Ahmed, Chris Garrett, Jeremy Faircloth, Chris Payne: ASP.Net Programming. Developer's Guide, Dreamtech, 2016.
2. Greg Buczek: ASP.Net Tips & Techniques, Tata McGraw Hill Edition, 2016.
3. Bolton, Justin Langford, Glenn Berry, Gavin Payne, Amit Banerjee, Rob Farley: Professional SQL Server 2019 internals and troubleshooting, Wiley India publication, October, 2019.

Web References:

1. http://msdn.microsoft.com/en-us/aa336522.aspx [For software download]
2. https://docs.microsoft.com/en-us/aspnet/overview [For ASP.NET technical documentation]
3. https://www.tutorialspoint.com/dotnet_core/dotnet_core_overview.htm [for .NET Core]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Students will be able to understand .NET framework, ASP.NET technologies and develop simple applications using the framework.
CO2 :	Students will be able to develop ASP .NET Web Form based application and get exposure to "How the web works?"
CO3 :	Students will be able to learn back-end Database Technology and how it can be integrated to Web Application or any Software.
CO4 :	Students will learn ASP .NET MVC framework for MVC Architecture.
CO5 :	Students will learn concepts of Routing, Filtering, Error Handling and will be able to develop MVC based Web Applications.
CO6 :	Students will be able to understand Web API implementation and its applications in real life scenario.

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes					
		CO1	CO2	CO3	CO4	CO5	CO6
1	Introduction to .NET framework and ASP.NET	√					
2	ASP .NET Web Forms		√				
3	ADO .NET (Database Connectivity)			√			
4	Introduction to MVC Architecture				√		
5	Application Development using MVC					√	
6	Introduction to Web Services						√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	3	-	2	1	-	-	2	-	-	2	1
CO2	2	3	2	2	-	2	1	-	-	2	2	1	3	1
CO3	1	2	2	-	-	-	-	-	-	-	2	1	3	2
CO4	2	3	3	-	2	-	-	-	-	-	2	1	3	2
CO5	3	2	3	2	-	-	3	-	-	-	-	2	3	3
CO6	3	2	3	-	-	-	2	-	-	-	-	1	2	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOC508 - Database Technologies

Pre-requisite: Basics of Database Management Systems and Data Structure

Methodology & Pedagogy: During theory lectures student will understand the essential concepts that act as the building blocks for many of the NoSQL products. It starts from the fundamentals of NoSQL and graduates to advanced concepts of architecture, storage internals and indexing. During Practical sessions, students will be required to design the applications using NoSQL Databases.

Outline of the course:

Unit Number	Title of the unit	Minimum number of hours	
		Theory	Practical
1	Introduction to Database System	05	36
2	Advanced Database Systems	07	
3	Introduction to NoSQL	06	
4	NoSQL Stores Data Models	06	
5	Storage Architecture	05	
6	Working and Querying (CRUD operation) with Data using MongoDB	07	

Total Hours (Theory): 36
Total Hours (Lab): 36
Total Hours: 72

Detail Syllabus:

Unit I: Introduction to Database System Hours 05

Basic Concepts: data, information, file, database, database systems, database management system Purpose and advantages of Database management system (over file systems), Applications of DBMS Various data models ER Model and Relational Model, Three level architecture, Structure of DBMS, Database actors and workers, Codd Rules

Unit II: Advanced Database Systems Hours 07

Basics of PL/SQL block, Usage of conditional statements and looping structures, Implicit and explicit cursor, User defined function and procedure, Introduction to row level and statement level triggers

Unit III: Introduction to NoSQL Hours 06

Introduction to SQL Relational Model, Overview and History of NoSQL Databases, Taxonomies by Data Model, Overview of No-SQL Storage: Key Value, Document-Oriented, Column-Family and Graphstorage. Techniques and Pattern: CAP Theorem, ACID Vs. BASE properties and Comparison of relational databases to new NoSQL stores.

Unit IV: NoSQL Stores Data Models Hours 06

Introduction, use and applications of: MongoDB, Cassandra, HBASE, Neo4j, Challenges NoSQL approach, Key-Value and Document Data Models, Column-Family Stores, Aggregate-Oriented Databases, Replication and Sharding, Overview of MapReduce on databases. Distribution Models, Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication.

Unit V: Storage Architecture Hours 05

Working with Document Databases, contrasting document databases with RDBMS, features of MongoDB, advantages of MongoDB over RDBMS, working with Column-Oriented Databases, Contrasting Column Databases with RDBMS, HBase Distributed Storage Architecture.

Unit VI: Working and Querying (CRUD operation) with Data using MongoDB Hours 07

Installing and running MongoDB on your System, use of JavaScript's shell in MongoDB, navigating your databases, viewing available data and collections, inserting data, querying data, using sort, limit and skip functions, retrieving single document, using aggregation commands, working with condition, updating data, upsert and save command, rename a collection

Core Books:

1. Ivan Bayross : SQL, PL/SQL The programming Language Oracle.
2. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Author: Sadalage, P. & Fowler, Publication: Pearson Education.

Reference Books:

1. Silberschatz, Korth, Sudarshan : Database System Concepts, 5th Edition, McGraw Hill.
2. Scoot Urban : Oracle 9i, PL/SQL Programming, Oracle Press.
3. S. K. Singh : Database Systems: Concepts, Design and Applications, Pearson Education
4. Definite Guide to MongoDB by Eelco Plugge, Peter Membrey and Tim Hawkins, Apress.
5. Getting started with NOSQL by Gaurav Vaish, PACKT.
6. Name: Redmond, E. &Wilson, Author: Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement Edition: 1st Edition.
7. NOSQL theory and Examples by Piotr fulmanski, Simple introduction series: e-book.
8. Guy Harrison, Next Generation Database: NoSQL and big data, Apress.

Web References:

1. http://docs.oracle.com/
2. http://plsql-tutorial.com/[For PL/SQL]
3. https://www.tutorialspoint.com/mongodb_for_absolute_beginners/index.asp [MongoDB For Absolute Beginners - Tutorialspoint]
4. https://www.tutorialspoint.com/neo4j/neo4j_data_model.htm [Neo4j - Data Model (tutorialspoint.com)]

Course Outcomes: Upon successful completion of the course, students will be,

CO1	Understand basic concepts regarding data, database systems and various data models.
CO2	Enhance the knowledge of the processes of Database Development and Administration using SQL and PL/SQL
CO3	Able to understand different types of NoSQL Databases
CO4	Architectures and common features of the basic types of NoSQL databases
CO5	Able to design Schema and implement CRUD operations, distributed data operations, and implement various column store internals

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to Database System	✓				
2	Advanced Database Systems		✓			
3	Introduction to NoSQL			✓		
4	NoSQL Stores Data Models			✓		
5	Storage Architecture				✓	
6	Working and Querying (CRUD operation) with Data using MongoDB				✓	✓

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	3	1	-	3	-	3	-	-	-	3	-
CO2	3	3	3	3	3	3	3	3	1	-	3	3	3	2
CO3	3	2	-	1	1	-	3	-	3	-	-	-	2	-
CO4	3	1	2	1	3	-	3	2	3	-	-	-	2	1
CO5	3	3	3	3	3	3	3	3	2	-	3	3	3	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

TEACHING SCHEME & DETAILED SYLLABUS

FOR

ONLINE MCA PROGRAMME

(Semester-II)

EFFECTIVE FROM ACADEMIC YEAR 2024-25

Teaching & Examination Scheme
Online Master of Computer Applications (M.C.A.) Programme
(Choice Based Credit System)

Effective from Year 2024 - 25
Semester-II

Course Code	Course Title	Teaching Scheme				Examination Scheme				
		Contact Hours			Credit	Theory		Practical		Total
		Theory	Practical	Total		CCE	SEE	CCE	SEE	
CAOE510	Python Web framework	3	3	6	6	30	70	30	70	200
CAOC513	Advanced Mobile Programming	3	3	6	6	30	70	30	70	200
CAOC514	Full-Stack Web Development	3	3	6	6	30	70	30	70	200
CAOC515	Software Engineering with Agile and DevOps	3	-	3	3	30	70	-	-	100
HSUA501	Academic Writing	-	3	3	3	-	-	30	70	100
		12	12	24	24	400		400		800

*CCE- Continuous and Comprehensive Evaluation (Assignments, Unit Tests, Sessional Test, Case Study, Attendance, Practical Test, Viva etc.) *SEE- Semester –End- Evaluation

Elective-II	
Course Code	**Course Title**
CAOE510	Python Web framework

CAOE510 - Python Web framework

Pre-requisite: Basics of Python Programming.

Methodology & Pedagogy: Theory sessions are required to address computational power of python through framework like DJango. The application of the topics taught in theory sessions is demonstrated in practical sessions. The case study will assist students in developing one working module in any paradigm using Python. The practical session should include actual application examples as well as MVT pattern integration.

Outline of the course:

Unit Number	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Introduction to Django	5	36
2	Django URL Patterns and Views	7	
3	Model and Database Setup for Django App	7	
4	Django Forms, Django & REST APIs	6	
5	Unit Testing with Django	6	
6	Admin, Security, Performance & Optimization	5	

Total Hours (Theory): 36
Total Hours (Lab) :36
Total Hours:72

Detail Syllabus:

Unit I: Introduction to Django **Hours 05**

Overview of Django, Python in context of Django, Features of Django, Model, View and Template pattern for Web Programming using Django, Installation guide for Django, First Application in Django.

Unit II: Django URL Patterns and Views **Hours 07**

Designing a good URL scheme. URLconfTricks, Streamlining Function, Imports, Using Multiple View Prefixes, Special-Casing URLs in Debug Mode, Using Named Groups, Understanding the Matching/Grouping Algorithm, Passing Extra Options to View Functions Using Default View Arguments, Special-Casing Views, Capturing Text in URLs, Determining What the URLconf Searches Against, Including Other URLconfs, How Captured Parameters Work with include (), How Extra URLconf Options Work with include().

Unit III: Model and Database Setup for Django App **Hours 07**

Defining Model using ORM approach, Relationship between Models, Model Inheritance, Meta inner model, Setting up database. Django with PostgreSQL, SQLite3, MySQL. Django without Database. Django Project with Development Server, Creating and Updating database using manage.py .

Unit IV: Django Forms, Django & REST APIs **Hours 06**

Form classes, Validation, Authentication, Advanced Forms processing techniques: Search forms, Creating Feedback form, Submission process of the form, Custom validation rules, Custom look and feel, Creating forms for models. REST APIs and its implementation. Django REST framework.

Unit V: Unit Testing with Django **Hours 06**

Overview / Refresher on Unit Testing and why it's good, Using Python's unittest2 library, Test, Test Databases, Doctests, Debugging Best Practices.

Unit VI: Admin, Security, Performance & Optimization **Hours 05**

Admin characterization in Django, Security facility in Django, Performance issues Django and Optimization in Django framework.

Core Books:

1. Adrian Holovaty, Jacob K. Moss: The Definitive Guide to Django:Web Development Done Right
2. Jeff Forcier, Paul Bissex, Wesley Chun: Python Web Development with Django, Pearson Education,2008.
3. Dusty Philips: Python 3 Object oriented Programming , PACKT publishing, 2010.

Reference Books:

1. Adrian Holovaty, Jacob Kaplan-Moss, et al.: The Django Book, 2013
2. Nathan Yergler: Effective Django, 2015
3. Steve Holden: Python Web Programming, 1st edition,2002.

Web References:

1. https://www.djangoproject.com/start/ [For Django Projects]
2. https://djangoforbeginners.com/initial-setup/ [For Django Setups]
3. https://docs.djangoproject.com/en/2.1/intro/tutorial01/ [For Django app]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Understand Django fundamentals and use its concepts to build and deploy robust web applications and apps.
CO2 :	Learn about Django URL patterns and views and deploy Django applications.
CO3 :	Learn how to configure Django for powerful databases and create the Django admin interface.
CO4 :	Learn about Django's security implications and how to create safe web applications with it.
CO5 :	To get some idea about unit testing concepts and its implementation through Django.

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to Django	√				
2	Django URL Patterns and Views		√			
3	Model and Database Setup for Django App			√		
4	Django Forms, Django & REST APIs				√	
5	Unit Testing with Django				√	
6	Admin, Security, Performance & Optimization					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	3	2	2	2	3	2	1	1	1	1	3	2
CO2	2	3	3	3	3	2	3	2	1	1	1	1	3	2
CO3	2	3	3	3	3	2	3	3	2	1	2	1	3	2
CO4	3	3	3	3	3	3	3	3	2	2	3	2	3	2
CO5	3	3	3	3	3	3	3	3	1	2	3	3	3	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOC513 - Advanced Mobile Programming

Pre-requisite: Object Oriented Programming concepts

Methodology & Pedagogy: During theory lectures illustrations emphasizing the need for basic features of Mobile Computing and Cross Platform- the Mobile Application Development platform will be given. During Practical sessions, students will be required to develop Mobile Application using Dart language in Flutter. Student shall also develop applications with elegant user interface that deal with data storage using Firebase and state management.

Outline of the course:

Unit Number	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Introduction to cross-platform development with Flutter and Dart	05	36
2	Creating UI with Flutter	07	
3	Building Apps with state	05	
4	Leveraging Flutter packages and structuring Flutter applications	08	
5	Incorporating backend data with Flutter application	05	
6	Integrating Flutter Application with Firebase and State Management	06	

Total Hours (Theory): 36
Total Hours (Lab): 36
Total Hours: 72

Detail Syllabus:

Unit I: Introduction to cross-platform development with Flutter and Dart **Hours 05**

Set up a new Flutter project using Android Studio. Introduction to Dart programming language: Libraries & imports, Built-in types, variables, operators, keywords, comments, Control Flow, Functions and parameters, Objectoriented programming in Dart.

Widget tree, Interface design: pre-made Flutter Widgets. Image and Text Widgets. App Icons for iOS and Android. Add and load image assets to Flutter projects. Run Flutter apps on iOS Simulator, Android Emulator and physical devices. Use the TextField Widget to take user input.

Unit II: Creating UI with Flutter **Hours 07**

Hot Reload and Hot Restart, Use of Pubspec.yaml file, custom assets and fonts. An introduction to the Widget build(), layout widgets: Columns, Rows, Containers and Cards. Material icons, Icons class. Theme widgets. Refactoring widgets. Dart annotations and modifiers. Immutability of Stateless and Stateful Widgets. Update screen with the build(). Custom Flutter Widgets.Difference between final and const in Dart. Maps, enums and the ternary operator. Functions and arguments in Dart. Multi-screen Flutter apps, routes and the Navigator widget. Flutter favours composition vs. inheritance (customizing widgets).

Unit III: Building Apps with state **Hours 05**

About Stateful and Stateless Widgets, callbacks. Declarative style of UI programming, Flutter widgets react to state changes. Import Dart libraries. Variables, data types and functions in Dart. Build flexible layouts. Understand the relationship between setState(), State objects and Stateful Widgets.

Unit IV: Leveraging Flutter packages and structuring Flutter applications **Hours 08**

Dart package manager, Flutter compatible packages. The structure of the pubspec.yaml file. Incorporate the audio players package to play sound. Functions in Dart and arrow syntax. Refactor widgets, Flutter's philosophy of UI as code. The lists and conditionals in Dart. Classes and objects. Understand Object Oriented Dart. Dart Constructors for Flutter widgets. Design patterns to structure Flutter apps. Structuring and organizing Flutter apps.

Unit V: Incorporating backend data with Flutter application Hours 05

Asynchronous programming in Dart and use of async/await. Stateful Widget lifecycle methods, Handling exceptions in dart. Null aware operators. Location data from both iOS and Android. Http package and live data from open APIs. Parse JSON data using the dart:convert library. State objects via the Stateful Widget. Use the TextField Widget to take user input. Pass data backwards using the Navigator widget.

Unit VI: Integrating Flutter Application with Firebase and State Management Hours 06

Hero animations in Flutter apps. Animation controller, custom animations. The use of Dart mixins. Firebase Cloud Firestore into your Flutter apps. Authentication with Firebase Auth package. A scrolling ListView, reusable elements. Dart Streams: listen data changes. StreamBuild. Manage state across the widget tree. Declarative vs. imperative programming. setState, prop drilling and lifting state up. todo list app. The BottomSheet, ListViewBuilder. The Flutter app architecture design patterns. Manage state with the Google recommended Provider package.

Core Books:

1. Marco L. Napoli: Beginning Flutter: A Hands On Guide to App Development: Wrox publication: 2019.

Reference Books:

1. Eric Windmill: Flutter in Action: Edition: Manning Publication: January 2020.
2. Alessandro Biessek: Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2: Packt publication: September 2019

Web References:

1. https://docs.flutter.dev/reference/tutorials
2. https://www.tutorialspoint.com/flutter/index.htm
3. https://www.javatpoint.com/flutter
4. https://fluttertutorial.in/

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	able to clear all object oriented programming and cross platform concepts
CO2 :	able to learn Flutter and Dart step by step
CO3 :	able to learn the reduce the code through native app performance, animated UI with material design and least testing
CO4 :	able to use Firebase to authenticate the users and use the remote database.
CO5 :	able to build engaging native mobile apps for both Android and iOS .

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to cross-platform development with Flutter and Dart	√				
2	Creating UI with Flutter	√	√			
3	Building Apps with state			√		
4	Leveraging Flutter packages and structuring Flutter applications			√		
5	Incorporating backend data with Flutter application					√
6	Integrating Flutter Application with Firebase and State Management				√	√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	3	2	1	3	-	1	-	1	-	-	1
CO2	3	1	3	2	3	1	2	1	1	-	1	1	1	2
CO3	3	-	3	1	3	1	1	2	1	-	1	1	1	2
CO4	3	3	3	3	3	1	1	3	1	-	2	2	3	1
CO5	3	3	3	1	3	1	1	3	1	1	2	2	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOC514 - Full-Stack Web Development

Pre-requisite: Working knowledge of HTML and JavaScript

Methodology & Pedagogy: During the theory sessions students will understands the various JavaScript frameworks and its architecture, also able to comprehend MEAN stack application concepts. During the practical sessions students will learn how to reduce the amount of code you write to build rich user interface applications, Modularizing code, retrieving data from back-end- server and manipulate it.

Outline of the course:

Unit No	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Introduction to JavaScript and JavaScript Frameworks.	04	36
2	Directives in AngularJS	07	
3	MVW-The AngularJS way	06	
4	Introduction to NodeJS	07	
5	Introduction to ExpressJS Framework	06	
6	Working with Database	06	

Total Hours (Theory): 36

Total Hours (Lab) :36

Total Hours:72

Detail Syllabus:

Unit I: Introduction to JavaScript Frameworks. **Hours 04**

Introduction JavaScript and its DOM concepts, Arrow function, JSON Objects, JavaScript Frameworks & Libraries, MEAN.JS introduction, Architecture of MEAN.JS.

Unit II: Introduction to AngularJS. **Hours 07**

Introduction to AngularJS, Features of AngularJS, Data Binding in angularJS, working with Expressions in angularJS, Introduction to Directives, Directive Lifecycle, Using Angular JS built-in directives, creating a custom directive, Working with $scope, $rootScope, Lifecycle of $scope.

Unit III: MVW-The Angular JS way. **Hours 06**

MVW Architecture: Model-View-Controller and Model-View-View-Model Architecture, Introduction to AngularJS Modules – Application module & Controller modules, Attaching Properties and functions to scope, Controller in external files, AngularJS Filters, working with Angular Forms, Form events, validating Angular forms.

Unit IV : Introduction to NodeJS **Hours 07**

Introduction to NodeJS, Features of NodeJS, Traditional web Server Model, Node.js process Model, Installation of NodeJS, working with Node Package Manager, Command Line options, NodeJS modules – Core Modules, Creating Local modules & Exporting Local modules, Third-party modules, Creating NodeJS web server.

Unit V : Introduction to ExpressJS Framework. **Hours 06**

Introduction to ExpressJS, Installing ExpressJS using NPM, Advantages of ExpressJS, Installing Express.js, building web server, configure routes, ExpressJS middleware's, Serve Static Resources using Express.js, working with HTTP methods of ExpressJS, ExpressJS routing, Creating RESTful API.

Unit VI : Working with Database. **Hours 06**

Introduction to Mongo DB, Installation of MongoDB – MongoDB Server & MongoDB Shell, MongoDB Compass, MongoDB Database, Collections, and documents, Basic operations using

MongoDB – Create, Insert, Update and Delete, Access MongoDB in Node.js, setting up mongoose, Connecting MongoDB, Insert, update and delete document.

Core Books:

1. Jeffry Houser : "Learn With: Angular 5, Bootstrap, and NodeJS", Kindle Edition,
2. Shyam Seshadri Brad Green:"AngularJS – Up and Running, Brad Green", Second Edition, O'REILLY
3. Basarat Ali Syed: "Beginning Node.js", Apress Publication.
4. Greg Lim : "Beginning MEAN Stack (MongoDB, Express, Angular, Node.js)", kindle Edition.

Reference Books:

1. Agus Kurniawan: "AngularJS Programming by Example 2017 Edition", Kindle Edition.
2. Adam Freeman: " Pro AngularJS 2017 Edition", Apress.
3. Krasimir Tsonev: "Node.js By Example", Packt Publishing
4. Ethan Brown: "Web Development with Node and Express", O'REILLY

Web References:

1. http://www.w3schools.com/angular/default.asp [Tutorial link for AngularJS]
2. http://www.tutorialspoint.com/angularjs/ [Tutorial link for AngularJS]
3. https://www.tutorialspoint.com/angularjs/angularjs_tutorial.pdf [E-book for AngularJS]
4. http://www.tutorialsteacher.com/nodejs/nodejs-modules

[Tutorial link for NodeJS]

5. https://www.javatpoint.com/mean-stack-tutorial

[MEAN stack development]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Familiar with JavaScript and JavaScript frameworks.
CO2 :	Able to understand basic fundamental of AngularJS and implement Model-View-View-Model Architecture.
CO3 :	Able to create NodeJS and ExpressJS applications.
CO4 :	Able to create Restful API to access data from MongoDB.
CO5 :	Able to create Single page application and MEAN stack application.

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to JavaScript and JavaScript Frameworks	√				√
2	Introduction to AngularJS		√			√
3	MVW-The Angular JS way		√			√
4	Introduction to NodeJS			√	√	√
5	Introduction to ExpressJS Framework			√	√	√
6	Working with Database				√	√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	-	-	-	-	2	-	1	-	-	-	-	1	-	-
CO2	1	2	2	1	2	-	1	1	-	1	1	2	1	2
CO3	1	-	3	1	3	-	2	1	-	-	1	2	1	1
CO4	2	2	3	2	3	2	2	3	-	2	2	3	3	3
CO5	2	3	3	2	3	2	2	3	2	2	3	3	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation

CAOC515 - Software Engineering with Agile and DevOps

Pre-requisite: Analysis and design of Information Systems, Fundamental concepts of Object Oriented Programming.

Methodology & Pedagogy:

During theory sessions, the focus will be given on different concepts of software engineering and related aspects such as Project management and Agile software development. The emphasis will be given on the basics of software engineering concepts and UML. Students will get exposure to various concepts of DevOps, as well as Process Improvement and Reengineering.

Outline of the course:

Unit Number	Title of the Unit	Minimum Number of Theory Hours
1	Introduction to Software and Software Engineering	5
2	Requirement analysis and specification	6
3	Requirement Modelling using UML	6
4	Essence of Agile Development	7
5	Role of DevOps in SE	7
6	Software maintenance and risk management	5

Total Hours (Theory): 36
Total Hours:36

Detail Syllabus:

Unit I: Introduction to Software and Software Engineering **Hours 05**

The Role of Software in today's era, Software Engineering: A Layered Technology/software process layers, Software Process Models, The Linear Sequential Model, The Prototyping Model, Evolutionary Process Models.

Introduction to software process: Generic process framework activities, Umbrella activities.

Unit II: Requirement analysis and specification **Hours 06**

Understanding to System and Software Requirement, types of software requirements, Requirement Specification (SRS), Requirement Analysis and Requirement Elicitation, Requirement Engineering.

Introduction to the basic concepts of Project Management Plan and Project Scheduling, Project scheduling principles, relationships and effort distribution, Process Definition and Tailoring, Process Database and Process Capability Baseline, Effort Estimation and Scheduling, Configuration Management.

Unit III: Requirement Modelling using UML **Hours 06**

Introduction to UML, Structure modelling: Class Diagram and Behavioural Modelling. Under Behavioural Modelling: Use case Diagram, Interaction Diagram, Activity Diagram.

Unit IV: Essence of Agile Development **Hours 07**

Introduction to Agile Software Development, Characteristics of Agile Process, Agile methods, Principles of Agile methods, Agile Process models: XP, ASD, DSDM, Scrum, Crystal, FDD, AM.

Unit V: Role of DevOps in SE **Hours 07**

Overview, Problem Case Definition, Benefits of Fixing Application Development Challenges, DevOps Adoption Approach through Assessment, Solution Dimensions, What is DevOps?, DevOps Importance and Benefits, DevOps Principles and Practices, 7 C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, How to Choose Right DevOps Tools, Challenges with DevOps Implementation, Must Do Things for DevOps, Mapping My App to DevOps Assessment, Definition, Implementation, Measure and Feedback.

Unit VI: Software maintenance and risk management **Hours 05**

Software risks, risk identification, risk projection, risk mitigation, monitoring and management. Introduction to software maintenance, types of maintenance.

Concept of software reengineering, Business Process Reengineering(BPR), BPR life cycle, reverse engineering, Applications of Software Engineering.

Core Books:

1. Roger S.Pressman, Software Engineering- A practitioner's Approach, McGraw-Hill International Editions
2. Grady Booch, James Rambaugh, Ivar Jacobson: The Unified Modelling Language User
3. Rajib Mall, Fundamentals of software Engineering, Prentice Hall of India.

Reference Books:

1. Ian Sommerville, Software engineering, Pearson education Asia
2. Pankaj Jalote, Software Engineering – A Precise Approach Wiley
3. Merlin Dorfman (Editor), Richard H. Thayer (Editor), Software Engineering
4. Robert C. "Uncle Bob" Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design
5. Deepak Gaikwad, Viral Thakkar, DevOps Tools from Practitioner's ViewPoint, Wiley

Web References:

1. https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Methodolog y [Introduction to Software Engineering/Process/Methodology]
2. https://www.uml-diagrams.org/ [UMLUnits]
3. https://nptel.ac.in/courses/106/101/106101061/ [Agile Software Development and Extreme Programming and Overview of DevOps]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Understanding Software Engineering Process Models
CO2 :	Able to analyze requirements and mange project
CO3 :	Get familiar with the concepts of Unified Modeling Language
CO4:	Understanding of the Agile software development concepts
CO5 :	Able to apply DevOps concepts and understand risk management

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to Software Engineering	√				
2	Requirement analysis and project management		√			
3	Understanding UML			√		
4	Agile Development				√	
5	DevOps					√
6	Software maintenance and risk management					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	3	2	2	3	3	2	2	2	1	2	3	3
CO2	3	2	2	2	2	3	3	3	3	2	3	2	3	3
CO3	2	2	3	2	2	2	2	2	2	1	1	2	3	2
CO4	3	3	3	2	2	2	2	1	1	1	3	2	3	2
CO5	2	2	3	2	1	3	2	1	1	1	2	1	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

TEACHING SCHEME & DETAILED SYLLABUS

FOR

Online MCA PROGRAMME

(Semester-III)

EFFECTIVE FROM ACADEMIC YEAR 2024-25

Teaching & Examination Scheme

Online Master of Computer Applications (M.C.A.) Programme

(Choice Based Credit System)

Effective from Year 2024 - 25
Semester-III

Course Code	Course Title	Teaching Scheme				Examination Scheme				
		Contact Hours			Credit	Theory		Practical		Total
		Theory	Practical	Total		Termwork	External	Termwork	External	
CAOE601 – CAOE604	Elective-III	3	3	6	6	30	70	30	70	200
CAOC605	Software Quality Assurance	3	3	6	6	30	70	30	70	200
CAOC606	Mini Project	-	12	12	12	-	-	120	280	400
CAOC607	Green Computing		2	2	2	-	-	30	70	100
		06	20	26	26	200		700		900

Elective-III	
Course Code	**Course Title**
CAOE601	Digital Image Processing
CAOE602	Object Oriented Programming for Smart Contracts
CAOE603	Data Analytics
CAOE604	Internet of Things

CAOE601 - Digital Image Processing
(200 Marks)

Contact Hours: 06

Pre-requisite: Basic knowledge of programming

Methodology & Pedagogy: The fundamental concepts of digital image processing should be covered in the beginning lectures. Various image enhancement and colour image processing techniques should be explained in details. The methods of image segmentation, representation and compressions should be taught in-depth.

Outline of the course:

Unit No	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Digital Image Fundamentals	05	36
2	Color Image Processing	07	
3	Image Enhancement	05	
4	Image Segmentation	07	
5	Image Representations and Recognition	06	
6	Image Compressions	06	

Total Hours (Theory): 36
Total Hours (Lab) : 36
Total Hours: 72

Detail Syllabus:

Unit I: Digital Image Fundamentals **Hours 05**

Introduction, Origin, Component of image processing system, Human visual system, Image as a 2D data, Image representation, Gray scale and Colour images, image sampling and quantization, Basic Relationships between Pixels.

Unit II: Colour Image Processing **Hours 07**

Colour fundamentals, Colour models, Colour transformation, Smoothing and Sharpening, Colour segmentation, Noise Models, Noise Reduction, Inverse Filtering.

Unit III: Image Enhancement **Hours 05**

Basic gray level Transformations, Histogram Processing, Spatial Filtering, Preliminary Concepts, Extension to functions of two variables, Image Smoothing, Image Sharpening, Frequency domain filtering, Homomorphic filtering

Unit IV: Image Segmentation **Hours 07**

Edge based segmentation, Region based segmentation, Region split and merge techniques, Region growing by pixel aggregation, optimal thresholding, Hough Transform, boundary detection.

Unit V: Image Representation and Recognition **Hours 06**

Boundary representation, Chain code, Polygonal Approximation, Signature, Boundary segments, Boundary description, Shape Number, Regional Descriptors, Topological Feature, Texture, Pattern and Patterns classes, Recognition Based on matching

Unit VI: Image Compression **Hours 06**

Introduction, Image compression model, Error free compression, Variable length coding, Bit plan coding, Lossless predictive coding, Lossy compression, Compression standards.

Core Books:

1. Rafael C. Gonzalez, Richard E. Woods: Digital Image Processing, Fourth edition, Pearson education, 2017
2. Anil K. Jain: Fundamentals of Digital Image Processing, First Edition, Pearson Education, 2015

Reference Books:

1. Burger, Wilhelm, Burge, Mark J: Principles of Digital Image Processing, Springer, 2009.
2. Milan Sonka, Vaclav Hlavac, Roger Boyle: Image Processing, Analysis, and Machine Vision, CL Engineering, 2007

Web References:

1. https://www.tutorialspoint.com/dip/ [Basics of Image Processing]
2. https://sisu.ut.ee/imageprocessing/book/1 [All concepts of digital image processing]
3. https://www.cs.auckIand.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic3.htm [Image segmentation]
4. http://www.eIeteI.p.Iodz.pI/mstrzeI/imageproc/enhancement1.PDF [Image enhancement]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Understand the concept of Explain various types of images and analyze various techniques for intensity transformation and spatial filtering with applications.
CO2 :	Understand the concepts of colour image processing and noise models.
CO3 :	Understand the concepts of image enhancement and its techniques.
CO4 :	Learn various image segmentation techniques.
CO5 :	Be able to represent and recognize images using various pattern recognition methods its compression techniques.

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Digital Image Fundamentals	√				
2	Colour Image Processing		√			
3	Image Enhancement			√		
4	Image Segmentation				√	
5	Image Representations and Recognition					√
6	Image Compressions					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	2	1	1	1	1	-	1	1	3	2	3
CO2	3	3	3	2	1	1	1	1	-	1	1	3	2	3
CO3	3	3	3	2	1	1	1	1	-	1	1	3	2	3
CO4	3	3	3	2	1	1	1	1	-	1	1	3	2	3
CO5	3	3	3	2	1	1	1	1	-	1	1	3	2	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOE602 - Object Oriented Programming for Smart Contracts
(200 Marks)

Contact Hours: 06

Pre-requisite: CAOE503: Blockchain Technology

Methodology & Pedagogy: During the lecture sessions, the students will learn about fundamental of smart contracts and related applications. Practical session will provide hands-on in order to implement smart contracts using Solidity programming language. Practical sessions will cover basics of Solidity, Expression and Control Statements, Object Oriented constructs and Testing, Debugging and Deployment. Through case study approach, students will explore smart contracts in real applications.

Outline of the course:

Unit Number	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Introduction to Smart Contracts	6	36
2	Basics of Solidity	7	
3	Expression & Control Statements	7	
4	OOPs Construct in Solidity	7	
5	Testing, Debugging and Deployment	5	
6	Smart contracts in Real World	4	

Total Hours (Theory): 36
Total Hours (Lab) : 36
Total Hours: 72

Detail Syllabus:

Unit I : Introduction to Smart Contracts **Hours 06**

An overview to the history of smart contracts, An intro to the life-cycle of a smart contract, Ethereum's smart contract languages, Interfacing with Ethereum Networks (overview of Ethereum Networks, Clients, Wallets, Transactions etc.), The Solidity Programming Language, Development Environments.

Unit II : Basics of Solidity **Hours 07**

Overview of Solidity, Solidity-Environment Setup, Basic Syntax, Smart contract layout, the structure of .sol source file, First Application, Understanding the different compiler versions and pragmas, Authoring smart contracts, Contract definitions, Basic data types, Variables, Variable Scope, Operators, Structs and Enums, Mapping and Arrays, Build-in Functions, User Functions.

Unit III : Expression & Control Statements **Hours 07**

Expressions and Control Structures, Valid expressions of the language, Conditional logic, Implementation of loops, Exception Handling, Events and Logging.

Unit IV : OOPs Construct in Solidity **Hours 07**

Contract constructor and self-destruct, Function Modifiers and Fallback functions, calling other contracts, Inheritance and Multiple Inheritance, Declaring Abstract Classes and Interfaces, Implementation of Abstract interfaces, Function Overloading .

Unit V : Testing, Debugging and Deployment **Hours 05**

Smart Contract testing, Basics of using Truffle for testing, types of testing in smart contract, Importance of testing in smart contracts, Debugging, Smart Contract Security – overview of attacks on Ethereum smart contracts Deployment.

Unit VI : Smart contracts in Real World **Hours 04**

Smart contracts in trading, investing, gaming, healthcare and real estate

Core Books:

1. Antonopoulos, Andreas M., and Gavin Wood: Mastering ethereum: building smart contracts and dapps : Edition (1st, 2nd , etc.) : O'Reilly Media: 2018.

Reference Books:

1. Ritesh Modi: Solidity Programming Essentials: A guide to building smart contracts and tokens using the widely used Solidity language: Edition (2nd) : Kindle Edition : Year of Publication.
2. Wei-Meng Lee: Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript: Apress: 2019

Web References:

1. https://www.blockchain-council.org/blockchain/smart-contracts-works/ [Smart Contracts Introduction]
2. https://101blockchains.com/solidity-tutorial/ [Solidity Introduction]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Able to understand the fundaments of Smart Contracts.
CO2 :	Able to learn basics and object oriented construct of Solidity programming language.
CO3 :	Able to implement smart contracts using Solidity programming language.
CO4 :	Able to test, debug and deploy smart contract.
CO5 :	Able to understand use cases of smart contract in real world.

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to Smart Contracts	√				
2	Basics of Solidity		√	√		
3	Expression & Control Statements		√	√		
4	OOPs Construct in Solidity		√	√		
5	Testing, Debugging and Deployment				√	
6	Smart contracts in Real World					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	1	1	2	-	1	3	-	-	-	-	2	1	3
CO2	3	1	2	1	3	-	3	-	-	-	-	2	2	3
CO3	3	1	2	1	3	-	3	-	-	-	-	2	2	3
CO4	3	1	3	2	3	-	3	-	-	-	-	2	3	3
CO5	1	1	1	2	1	-	3	-	-	2	2	2	2	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOE603 - Data Analytics
(200 Marks)

Contact Hours: 06

Pre-requisite: CAOC508: Database Technologies

Methodology & Pedagogy: During theory lectures the emphasis will be given on the basics of data analytics and related tools and techniques. Students will be introduced to the concepts of data science, exploratory data analysis, supervised and unsupervised learning methods. Applications as well as research trends and future direction of data analytics will be discussed with the length. During the practical sessions, students will be introduced to tools of data analytics such as R and Python. Students will be given appropriate case studies of data analytics to get the real time exposure of data analytics.

Outline of the course:

Unit No	Title of the Unit	Minimum Number of Theory Hours	Practical
1	Introduction of Data Analytics	06	36
2	Exploratory Data Analysis	06	
3	Supervised & Unsupervised Learnin Methods	06	
4	Deep Re-enforcement Learning	06	
5	Data visualization	06	
6	Recent trends and future directions	06	

Total Hours (Theory): 36
Total Hours (Lab): 36
Total Hours: 72

Detail Syllabus:

Unit I : Introduction of Data Analytics Hours 06

What is Data Science? ,Big Data and Data Science hype ,Why now? , Datafication , Current landscape of perspectives , Skill sets needed, Data Analysis cycle, Data analysis Vs Data analytics, Life cycle of data.

Unit II : Exploratory Data Analysis Hours 06

Philosophy of EDA - The Data Science Process ,Statistical Inference , Populations and samples , Statistical modeling, probability distributions, fitting a model Introduction to R ,Basic tools (plots, graphs and summary statistics) of EDA, SQL with data life cycle, Python for EDA.

Unit III : Supervised & Unsupervised Learning Methods Hours 06

Supervised methods: Linear Regression , Classification Trees, Random Forest, Clustering, Support Vector Machine, Regularization, Radial Basis, Association Rule Mining, Dimensionality Reduction - Singular Value Decomposition – Principal Component Analysis

Unit IV : Deep Re-enforcement Learning Hours 06

Introduction to Neurons, Activation functions, single and multi-layer perceptron, back propagation, Application, Hyper-parameters and Estimation, Gradient Descent, Curiosity driven learning.

Unit V : Data visualization Hours 06

Introduction of data visualization, Power BI, and Tableau, ggplot+ggplot2, Seaborn.

Unit VI : Recent trends and future directions Hours 06

Data Visualization - Basic principles, ideas and tools for data visualization, Data Science and Ethical Issues - Discussions on privacy, security, ethics , Next-generation data scientists, Basics of Big Data analytics,

Core Books:

1. Cathy O'Neil and Rachel Schutt: Doing Data Science, Straight Talk From The Frontline, O'Reilly. 2014.
2. Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman,Mining of Massive Datasets Cambridge University Press,2nd Edition, New York, NY, USA,2014.
3. Howard B. Demuth, Mark H. Beale, Orlando De Jess, and Martin T. Hagan, Neural Network Design , paperback USA, 2nd Edition,2014.

Reference Books:

1. Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. , Probability & statistics for engineers & scientists ,9th edition, Prentice Hall,2012.
2. Haykin, S. S., Haykin, S. S., Haykin, S. S., & Haykin, S. S., Neural networks and learning machines Pearson, Vlolume 3,2009.
3. Mohammed J. Zaki and Wagner Miera Jr, Data Mining and Analysis: Fundamental Concepts and Algorithms.,Cambridge University Press. 2014.

Web References:

1. https://onlinecourses.nptel.ac.in/noc17_mg24/preview [Online Data Analytics Course]
2. https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm [Exploratory Data AnalysisMaterial]
3. https://datahoarder.io/Humble%20Bundle%20Books/Humble%20Book%20Bundle_%20Data%20Science%20presented%20by%20O_Reilly/doingdatascience.pdf [Data Science E-Book]
4. http://www.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf [Supervised andunsupervised methods tutorials]
5. https://yourstory.com/2017/12/data-analytics-future-trends/ [Future Trends of Data Analytics]
6. http://www.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf [Supervised andunsupervised methods tutorials]
7. https://yourstory.com/2017/12/data-analytics-future-trends/ [Future Trends of Data Analytics]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Describe what Data Science and Data Analytics are and the skill sets needed to be a data scientist.
CO2 :	Understand significance of exploratory data analysis in statistical and visualization aspects.
CO3 :	Understand and apply data analytics techniques such as supervised, unsupervised and EDA.
CO4 :	Understand the importance of self-learning environments
CO5 :	Able to understand the recent trends and future directions of data analytics.

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction of Data Analytics	√				
2	Exploratory Data Analysis		√			
3	Supervised & Unsupervised Learning Methods			√		
4	Deep Re-enforcement Learning				√	
5	Data visualization				√	
6	Recent trends and future directions					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	1	-	-	-	1	-	-	-	2	3	1
CO2	2	2	1	1	-	1	-	-	-	-	-	2	2	1
CO3	1	3	3	2	-	1	3	-	-	-	-	2	1	1
CO4	2	1	2	2	2	1	2	-	-	-	-	2	1	2
CO5	1	1	2	3	1	-	2	-	-	-	-	3	2	1

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOE604 - Internet of Things

(200 Marks)

Contact Hours: 06

Pre-requisite: - None

Methodology & Pedagogy: In order to achieve the objectives and goals, students will be taught the basics of IOT with its structure, layers and applications. Fundamentals controllers are introduced which can be used to implement the IOT based projects. Students will also be introduced to interface GPIO with controllers like arduino and Raspberry PI. They can also learn to develop a desktop application and mobile application which can control the device remotely.

Outline of the course:

Unit No	Title of the Unit	Minimum Number of Hours	
		Theory	Practical
1	Introduction to Internet of Things	06	36
2	Programming Internet of Things System	08	
3	Sensors and Actuators for Internet of Things System	04	
4	Designing Mobile Application and Webpage for Internet of Things System	06	
5	ESP8266 and ESP32 Microcontroller for IOT	06	
6	Raspberry Pi for IOT Projects	06	

Total Hours (Theory): 36
Total Hours (Lab) :36
Total Hours:72

Detail Syllabus:

Unit I : Introduction to Internet of Things : Hours: 06

Introduction - Overview of Internet of Things (IoT), The building blocks of IoT, Various technologies making up IoT ecosystem, IoT levels, IoT design methodology, The Physical Design and Logical Design of IoT, Functional blocks of IoT and Communication Models, Development Tools used in IoT.

Unit II : Programming Internet of Things System: Hours: 08

Arduino / node MCU controller for IOT Programming, Embedded C data types, variables, constants and operators, programming using control statements, loops, string and array, Arduino IDE. Variables and Numbers, Looping Structures, Conditional Statements, Lists, Tuples and Dictionaries, Type Conversions, Function declaration, calling functions and passing values, Function Returning values. Exiting from functions.

Unit III : Sensors and Actuators for Internet of Things System: Hours: 04

Interfacing with various sensors like temperature sensor, PIR sensor and ultrasonic sensor. Interfacing with servomotor, DC motor, Gas sensor.

Unit IV : Designing Mobile Application and Webpage for Internet of Things System Hours: 06

Basic concepts of mobile application development, designing webpage for IoT user, Connecting App / webpage with cloud and controller as per Internet of Things system levels.

Unit V : ESP8266 and ESP32 Microcontroller for IOT Hours: 06

Basics of Wireless Networking, Introduction to ESP8266 Wi-Fi Module, Various Wi-Fi library, Web server- introduction, installation, configuration, Posting sensor(s) data to web server, Introduction to ESP32 microcontroller.

Unit VI : Raspberry Pi for IOT Projects Hours: 06

Introduction to Raspberry Pi, Why Raspberry pi, Features of Raspberry pi, Basic set up and first boot configuration, Different uses of Raspberry pi, Different Versions of Raspberry pi, Vision of Raspberry pi, Different components of the Board.

Core Books:

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach, VPT Publication, 2014.
2. Dr. Simon Monk: Programming the Raspberry Pi: Getting Started with Python, McGraw Hill Publication

Reference Books:

1. M. Richardson, S. Wallace: Getting Started with Raspberry Pi, O'Reilly, 2012
2. Dr. Simon Monk: Programming the Raspberry Pi: Getting Started with Python, McGraw Hill Publication

Web References:

1. https://www.tutorialspoint.com/arduino/index.htm [Arduino Fundamentals]
2. www.tinkercad.com [Arduino Simulator]
3. https://www.youtube.com/watch?v=UUOChoCbty8[Raspberry Pi - How to start programming with Python]
4. https://www.raspberrypi.org/documentation/usage/python/[Step by Step Programming]

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Ability to understand IOT Terminology
CO2 :	Understanding IOT Programming
CO3 :	Understanding use of sensors and actuators
CO4 :	Learning Mobile app for IOT Projects
CO5 :	Designing IOT projects using ESP8266 and ESP32 and Raspberry PI

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to Internet of Things	√				
2	Programming Internet of Things System		√			
3	Sensors and Actuators for Internet of Things System			√		
4	Designing Mobile Application and Webpage for Internet of Things System				√	
5	ESP8266 and ESP32 Microcontroller for IOT					√
6	Raspberry Pi for IOT Projects					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	1	3	1	1	1	1	1	1	-	1	-	-
CO2	3	1	2	3	1	1	1	2	2	-	-	1	1	-
CO3	2	2	2	1	1	1	2	2	2	-	-	2	1	1
CO4	2	2	3	2	1	2	2	2	2	-	-	2	1	1
CO5	1	2	3	2	1	2	3	2	2	-	-	2	1	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOC605 - Software Quality Assurance
(200 marks)

Contact Hours: 06

Pre-requisite: None

Methodology & Pedagogy: The theory sessions will cover the concept of software testing that can be applied to achieve quality of software. The practical sessions will cover the application of testing techniques at various levels of software development using manual and automated testing tools.

Outline of the Course:

Unit No.	Title of the Unit	Minimum Numbers of Hours	
		Theory	Practical
1	Introduction to software quality and testing process	06	36
2	Unit level testing	06	
3	Integration testing	06	
4	System Test Design, Planning & Automation	07	
5	System Test Execution & Acceptance Testing	06	
6	Software Quality Assurance	05	

Total Hours (Theory): 36
Total Hours(Lab): 36
Total Hours: 72

Detailed Syllabus:

Unit – I: Introduction to software quality and testing process Hours: 06

Concept of Software Quality, Role of testing, Design test case: Stateless and State oriented test cases sources of Information for Test Case Selection, verification and validation, types of software testing, smoke testing, sanity testing, regression testing, re-testing, performance testing, security testing, compatibility testing, recovery testing, brief outline of levels of testing.

Unit – II: Unit level testing Hours: 06

White box testing methods, concept of unit level testing, Introduction to Test case design techniques, Static Techniques: Informal Reviews, Walkthroughs, Technical Reviews, Inspection. Introduction to Dynamic Techniques. Control flow testing: Statement Coverage Testing, Branch Coverage Testing, Path Coverage Testing, Conditional Coverage Testing, Finite state machine(FSM) : Handling events and state transitions, Guard conditions, activities state, types of states, test coverage criteria.

Unit – III: Integration testing Hours: 06

Concept of Integration Testing, System Integration Techniques, Software and Hardware Integration, Test Plan for System Integration, Off-the-Shelf Component Integration, System level testing, types of system level tests, Black box testing methods: Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing, Error Guessing, Category Partition.

Unit - IV: System Test Design, Planning & Automation Hours: 07

Test Design Factors, Requirement Identification, Characteristics of Testable Requirements, Test Design Preparedness Metrics, Test Case Design Effectiveness Structure of a System Test Plan, Introduction and Feature Description, Assumptions, Test Approach, Test Suite Structure, Test Environment, Test Execution Strategy, Test Effort Estimation, Scheduling and Test Milestones, System Test Automation, Evaluation and Selection of Test Automation Tools, Test Selection Guidelines for Automation, Characteristics of Automated Test Cases, Structure of an Automated Test Case, Test Automation Infrastructure, , Entry criteria and exit criteria for system test plan , design in software testing life cycle.

Unit - V: System Test Execution & Acceptance Testing Hours: 06

Preparedness to Start System Testing, Metrics for Tracking System Test, Metrics for Monitoring Test Execution, Beta Testing, First Customer Shipment, System Test Report, Product Sustaining, Measuring Test Effectiveness. Types of Acceptance Testing, Acceptance Criteria, Selection of Acceptance Criteria, Acceptance Test Plan, Acceptance Test Execution, Acceptance Test Report, Acceptance Testing in extreme Programming, Entry criteria and exit criteria for system test execution and test closure in software testing life cycle.

Unit - VI: Software Quality Assurance Hours: 05

Five Views of Software Quality, McCall's Quality Factors and Criteria, Quality Factors Quality Criteria, Relationship between Quality Factors and Criteria, Quality Metrics, Quality Characteristics, Software Quality ISO Standards. Elements of Software Quality Assurance, SQA Task, Goals and Metrics, Formal approaches to SQA, Statistical Software Quality Assurance, Software Reliability, SQA Plan.

Core Books:

1. Sagar Naik, Piyu Tripathy: Software Testing and Quality Assurance, Theory and Practice, Wiley, 2008.

2. Roger S Pressman: Software Engineering – A Practitioner's Approach, 7th Edition, McGRAW HILL International Edition, 2010.

3.Lee Copeland: A Practitioner's Guide to Software Test Design, Artech House Publishers,2004.

Reference Books:

1. Boris Beizer: Software Testing Techniques: 2nd Edition, Van Nostrand Reinhold, 1990.

2. Daniel Galin: Software Quality Assurance, Pearson Education.

3. Ron Patton: Software Testing, Pearson Education, 2001.

Web References:

1. https://www.geeksforgeeks.org/software-engineering-white-box-testing/?ref=lbp [Introduction to Software Testing]

2. https://www.javatpoint.com/software-development-life-cycle [Software Development Life Cycle]

3. https://www.cs.ccu.edu.tw/~naiwei/cs5812/st7.pdf [Finite State Machine Testing]

Course Outcomes: Upon successful completion of the course, students will be

CO1 :	Able to understand basic concepts of software quality and testing
CO2 :	Familiar with concept of unit testing and methods of white box testing
CO3 :	Able to apply black box testing methods
CO4 :	Familiar with different concepts of software testing techniques
CO5 :	Able to understand various standards of quality

Course Outcomes Mapping:

Unit No.	Unit Name	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1	Introduction to software quality and testing process	√				
2	Unit level testing		√			
3	Integration testing			√		
4	System Test Design, Planning & Automation				√	
5	System Test Execution & Acceptance Testing				√	
6	Software Quality Assurance					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	3	2	2	3	3	2	2	2	1	2	3	3
CO2	3	2	2	2	2	3	3	3	3	2	3	2	3	3
CO3	2	2	3	2	2	2	2	2	2	3	1	2	2	2
CO4	3	3	3	2	3	3	3	1	3	3	3	2	2	1
CO5	3	2	3	2	1	3	2	2	2	2	2	1	2	1

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

CAOC606 - Mini Project

(400 Marks)

Contact Hours: 12

Objective:
The main objective of this mini project is to let the students apply the programming knowledge to a real-world situation/problem and exposed the students with specific programming skills and help in developing a working model in terms of application.

Course Outcomes:

C01:	Student will understand the implementation of concepts of SDLC and Software Engineering.
C02:	The programming concepts they learn during their academics; it will be converted in to the actual implementations.
C03:	Students will be exposed to understand the requirement of proposed software and implement these requirements in terms of programming logic and methods.
C04:	Students must understand the difference between a program and professional application/product/software.
C05:	Students will learn different categories of applications like Desktop application, Web applications, etc.

Guidelines:
Mini Project is in house project development. Every student is required to carry out Mini Project work under the supervision of a guide provided by the placement Coordinator. The guide shall monitor progress of the student continuously. A candidate is required to present the progress of the Mini Project work during the semester as per the schedule provided by the placement Coordinator.
Mini Project proposal should be prepared in consultation with project guide. It should clearly state the objectives and environment of proposed Mini Project to be undertaken. Project documentation must be with the respect to the project only. Project report should strictly follow the points suggested in format of project report. Placement coordinator will provide the format of project report. Student has to submit one copy of Mini Project to the institute. Each Student is required to make a copy of Mini Project in CD and submit along with Mini Project report.

Evaluation:

The project must be evaluated in two aspects:

a. Internal (100 Marks):
   i. Reporting to internal project guide
   ii. Incorporation of suggestions by project guide
   iii. Internal Project viva examination

b. External (300 Marks):
   i. Project Report Preparation & Evaluation
   ii. External Project Viva Examination

Course Code	Course Title	Teaching Scheme		Internal	End Semester Examination		Total
		Contact Hours	Credit	Continuous Evaluation	Report	Presentation & Viva	
CAUC606	Mini Project	12	12	100	100	200	400

Web References:

1. http://techwhirl.com/writing-software-requirements-specifications/[For effective SRS]
2. http://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html [For best practices of Software Project Development]
3. http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf [Requirement analysis guidelines]
4. http://www.cs.wustl.edu/~schmidt/PDF/design-principles4.pdf[Software Design Principles and Guidelines]
5. http://www.cse.hcmut.edu.vn/~hiep/KiemthuPhanmem/Tailieuthamkhao/Effective%20Software%2 0Testing%20-2050%20specific%20ways%20to%20improve%20your%20testing.pdf[ForEffective Software Testing]
6. http://www.cs.uics.edu/~jbell/CourseNotes/OO_SoftwareEngineering/SE_Project_Report_Template .pdf[For guidelines to prepare software project report]

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	2	3	3	3	3	3	2	2	2	3	2
CO2	3	3	3	2	3	3	3	3	3	2	2	2	3	2
CO3	3	3	3	2	3	3	3	3	3	2	2	2	3	2
CO4	3	3	3	2	3	3	3	3	3	2	2	2	3	2
CO5	3	3	3	2	3	3	3	3	3	2	2	2	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation

CAOC607 - Green Computing
(100 Marks)

Contact Hours: 02

Pre-requisite: None

Methodology & Pedagogy: During practical sessions, the basic concepts regarding green computing, green framework and green compliance will be discussed. The core concepts towards green computing will be covered so that students can analyze and understand various aspects related with green computing.

Outline of the course:

Week No	Practical	Description
1 - 2	**Carbon footprint of Green Computing**	• Analysis and survey about the carbon footprint of campus. • Measuring the carbon footprint, a small MIDC Company. • Carbon footprint for FMCG companies • CORPORATE CARBON FOOTPRINT AND CARBON NEUTRALITY. • Measures to reduce carbon footprint
3 - 4	**Energy Conservation in Green Computing**	• Study on Green Assets (Buildings, Data Centers, Networks, and Devices, Green Information Systems) Design and Development Models. • Plan to Cut Down Your Electricity Bill • Measuring and monitoring use of Power in XXX Company.

		• A survey of steps taken to be energy efficient by Malls / Shopping centers • Energy Efficient cooling solutions for homes and offices • Electricity Smart Grids and smart energy systems
5 - 6	**Recycling in Green Computing**	• Recycling of IT waste in Colleges • Recycling vs Reuse • Recycling initiatives taken up by XXX Housing Society: A Case Study Damage caused by improper recycling of e-waste in developing countries like India or China
7 - 8	**Going Paperless and Datacenters in Green Computing**	• A Case study of a traditional company going paperless with the use of Electronic media • Going paperless in Government Departments • Challenges in going paperless in Indian Context • Economic benefits of going Paperless • Survey of best energy-efficient practices in data centers around the world • Designing a datacentre with use of green technology. • Design considerations of datacentres for efficient cooling • Impact of Datacenters on the environment.

9-10	**Review of Green Initiatives in India and abroad**	• Submitting a small research study on National Mission for a Green India (GIM). • Submitting a small research study on Indian Green Building Council (IGBC). • Submitting a small research study on The Indian Council of Agricultural Research. • Submitting a small research study on LEED INDIA. • Basel Convention on the Control of Transboundary Movements of Hazardous Waste and their Disposal. • Initiatives taken by Europe to reduce the toxic effect of e-waste on environment • E-waste management rules implemented by Ministry of Electronics & Information Technology by Government of India.
11-12	**Creating Prototype model**	• Build a prototype/Working model of a wind turbine at home. • Build a prototype/Working model of a simple solar equipment's. • Build a prototype/Working model of Water Purification System. • Rain water harvesting model preparation. • Creating greywater recycling system. Paper bag making and distributing.

Total Hours (Practical): 24
Total Hours:24

Core Books:

1. Bhuvan Unhelkar: "Green IT Strategies and Applications-Using Environmental Intelligence", CRC Press, June 2014.
2. Woody Leonhard, Katherine Murray: "Green Home computing for dummies", August 2012.

Reference Books:

1. Alin Gales, Michael Schaefer, Mike Ebbers: "Green Data Center: steps for the Journey", Shroff/IBM rebook, 2011.
2. John Lamb: "The Greening of IT", Pearson Education, 2009.
3. Jason Harris: "Green Computing and Green IT- Best Practices on regulations & industry", Lulu.com, 2008.
4. Carl speshocky: "Empowering Green Initiatives with IT", John Wiley & Sons, 2010.
5. Wu Chun Feng : "Green computing: Large Scale energy efficiency", CRC Press

Web References:

1. https://dataconomy.com/2022/06/what-is-green-computing-definition/
2. https://www.techrockstars.com/the-essential-guide/what-is-green-computing/
3. https://www.smallbusinesscomputing.com/guides/green-computing/

Course Outcomes: Upon successful completion of the course, students will be,

CO1 :	Can understand the fundamentals and Carbon footprint of Green Computing
CO2 :	Can analyze and understand Green assets and its importance
CO3 :	Can analyze the Green computing Grid Framework.
CO4 :	Can understand the issues related with Green compliance.
CO5 :	Can study and develop various applications and case studies.

Course Outcomes Mapping:

Week No.	Practical Title	Course Outcomes				
		CO1	CO2	CO3	CO4	CO5
1-2	Green IT Fundamentals and Carbon footprint of Green Computing	√				
3-4	Green Assets and Energy Conservation in Green Computing		√			
5-6	Recycling and conservation		√			
7-8	Going Paperless and Datacenters, Green computing frameworks			√		
9-10	Review of Green Initiatives in India and abroad				√	
11-12	Green Modeling, prototype and applications					√

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
CO2	-	2	-	-	-	2	-	-	-	-	2	-	2	2
CO3	-	2	2	2	-	2	2	-	-	2	2	-	-	2
CO4	-	-	-	-	2	3	-	-	-	-	-	-	-	2
CO5	-	-	-	-	-	1	-	-	1	1	2	-	-	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation "-"

TEACHING SCHEME & DETAILED SYLLABUS

FOR

Online MCA PROGRAMME

(Semester-IV)

EFFECTIVE FROM ACADEMIC YEAR 2024-25

Teaching & Examination Scheme
Online Master of Computer Applications (M.C.A.) Programme
(Choice Based Credit System)

Effective from Year 2024 - 25
Semester-IV

Course Code	Course Title	Teaching Scheme				Termwork	End Semester Examination		Total
		Contact Hours			Credit	Continuous Evaluation	Report	Presentation & Viva	
		Inst.	Industry	Total					
CAOC608	Dissertation / Project Work	2	28	30	30	240	280	280	800

CAOC608 – Dissertation / Project Work

(800 Marks)

Contact Hours: 30

Objective:
The main objective of this Dissertation is to let the students apply the programming knowledge to a real-world situation/problem and exposed the students with specific programming skills and help in developing a working model in terms of application.

Course Outcomes:

CO1:	Student will understand the implementation of concepts of SDLC and Software Engineering.
CO2:	The programming concepts they learn during their academics; it will be converted in to the actual implementations.
CO3:	Students will be exposed to understand the requirement of proposed software and implement these requirements in terms of programming logic and methods.
CO4:	Students must understand the difference between a program and professional application/product/software.
CO5:	Students will learn different categories of applications like Desktop application, Web applications, etc.

Guidelines:
Dissertation is in house project development. Every student is required to carry out Dissertation work under the supervision of a guide provided by the placement Coordinator. The guide shall monitor progress of the student continuously. A candidate is required to present the progress of the Dissertation work during the semester as per the schedule provided by the placement Coordinator.
Dissertation proposal should be prepared in consultation with project guide. It should clearly state the objectives and environment of proposed Dissertation to be undertaken. Project documentation must be with the respect to the project only. Project report should strictly follow the points suggested in format of project report. Placement coordinator will provide the format of project report. Student has to submit one copy of Dissertation to the institute. Each Student is required to make a copy of Dissertation in CD and submit along with Dissertation report.

Evaluation:

The project must be evaluated in two aspects:

a. Internal (200 Marks):
   i. Reporting to internal project guide
   ii. Incorporation of suggestions by project guide
   iii. Internal Project viva examination

b. External (600 Marks):
   i. Project Report Preparation & Evaluation
   ii. External Project Viva Examination

Course Code	Course Title	Teaching Scheme		Internal	End Semester Examination		Total
		Contact Hours	Credit	Continuous Evaluation	Report	Presentation & Viva	
CAUC608	Dissertation / Project	30	30	240	280	280	800

Web References:

1. http://techwhirl.com/writing-software-requirements-specifications/[For effective SRS]
2. http://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html [For best practices of Software Project Development]
3. http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf [Requirement analysis guidelines]
4. http://www.cs.wustl.edu/~schmidt/PDF/design-principles4.pdf[Software Design Principles and Guidelines]
5. http://www.cse.hcmut.edu.vn/~hiep/KiemthuPhanmem/Tailieuthamkhao/Effective%20Softwar %2 0Testing%202050%20specific%20ways%20to%20improve%20your%20testing.pdf [ForEffective Software Testing]
6. http://www.cs.uics.edu/~jbell/CourseNotes/OO_SoftwareEngineering/SE_Project_Report_Template .pdf[For guidelines to prepare software project report]

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	3	3	3	3	3	2	2	2	3	2	-
CO2	3	3	3	2	3	3	3	3	3	2	2	2	3	2	-
CO3	3	3	3	2	3	3	3	3	3	2	2	2	3	2	-
CO4	3	3	3	2	3	3	3	3	3	2	2	2	3	2	-
CO5	3	3	3	2	3	3	3	3	3	2	2	2	3	2	-

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation