ಬಾಗಲಕೋಟ ವಿಶ್ವವಿದ್ಯಾಲಯ
(ಕರ್ನಾಟಕ ಸರ್ಕಾರದ ವಿಶ್ವವಿದ್ಯಾಲಯ)
Bagalkot University
(A State Public University of Govt.of Karnataka)

BAGALKOT UNIVERSITY,
JAMKHANDI

SCHOOL OF BASIC SCIENCES

Department of COMPUTER APPLICATION

for

MASTER OF COMPUTER APPLICATION

REGULATIONS
&
SCHEME OF EXAMINATION

As per
CHOICE BASED CREDIT SYSTEM (CBCS)

With effect from Academic Year 2024-25

Syllabus of III Semester MCA programme,
CHOICE BASED CREDIT SYSTEM(CBCS)
(According new regulations w.e.f. 2024-25)

Semester-III	III SEMSTER MCA w.e.f.2024-25								
	Course	Subject Name	Teaching Hrs per week	Practical Hrs/ week	Examination				Credits
					Duration (Hrs.)	Marks			
						Theory/ Practical	IA	Total	
Core Subject	126MCA03XXXCSC08T	Software Engineering	4	--	3	80	20	100	4
	126MCA03XXXCSC09T	Web Programming	4	--	3	80	20	100	4
	126MCA03XXXCSC10T	Programming using Python	4	--	3	80	20	100	4
	126MCA03XXCSC05L	Web Programming Lab	--	4	3	80	20	100	3
	126MCA03XXXCSC06L	Python Programming-Lab	--	4	3	80	20	100	3
Soft Core / Specialization/ Optional	126MCA03XXXSCC03T	Artificial Intelligence	4	--	3	80	20	100	4
Core Elective	126MCA03XXXCSC11T	a. Cyber Forensic & Security	4	--	3	80	20	100	4
	126MCA03XXXCSC12T	b. Internet of Things	4	--	3	80	20	100	4
	126MCA03XXXCSC13T	c. Pattern Recognition	4	--	3	80	20	100	4
	126MCA03XXXCSC14T	d. Embedded Systems	4	--	3	80	20	100	4
	126MCA03XXXCSC15T	e. Cloud Computing	4	--	3	80	20	100	4
	Total		20	8				700	26

CS: Core Subject SC: Soft Core PL: Practical CE: Core Elective Course

126MCA03XXXCSC08T: Software Engineering

Teaching: 4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT I **10 Hrs**

Introduction to Software Engineering: FAQs about software engineering, systems engineering, system availability and reliability, software processes, project management.

UNIT II **10 Hrs**

Software Requirement: Software requirements, requirements engineering project, system models, critical systems specification, formal specification.

UNIT III **12 Hrs**

Software Design: Architectural designs, distributed system architectures, application architectures, object oriented design, real-time software design, user interface design.

UNIT IV **10 Hrs**

Software Development: Rapid software development, software reuse, component-based software engineering, critical systems development, software evolution.

UNIT V **10 Hrs**

Verification, Validation and Management: Software inspections, static analysis, verification and formal methods, software testing, critical systems validation.
Managing people, software cost estimation, quality management, process improvement, configuration management.

References:

1. Sommerville, Software Engineering, 8/e, Pearson Education.
2. Pressman S. Roger, Software Engineering, Tata McGraw Hill.
3. JalotePankaj, An integrated Approach to Software Engineering, Narosa Publishing House.
4. Shooman, Software Engineering, McGraw Hill.
5. C. Ghezzi, M. Jazayeri and D. Mandrioli, Fundamentals of Software Engineering, Prentice Hall of India

126MCA03XXXCSC09T: Web Programming

Teaching: 4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT I **10 Hrs**

Overview: Web page Designing using HTML, Java Script-Object, names, literals, operators and expressions- statements and features-events - windows - documents - frames - data types - built-in functions- Browser object model - Verifying forms.-HTML5- CSS3- HTML 5 canvas. XML: DTD, Namespaces, XML schemas, displaying raw XML documents, Displaying XML documents with CSS, XSLT style sheets, XML processors

UNIT II **10 Hrs**

PHP : Server-side web scripting, Installing PHP, Adding PHP to HTML, Syntax and Variables, Passing information between pages, Strings, Arrays and Array Functions, Numbers, Basic PHP errors / problems. Database access with PHP and MySQL, PHP/MySQL Functions, Displaying queries in tables, Building Forms from queries, String and Regular Expressions, Sessions, Cookies and HTTP, Type and Type Conversions.

UNIT III **12 Hr**

Ruby on Rails: Origins and uses of Ruby, Scalar types and their operations, Simple input and output, Control statements, Arrays, Hashes, Methods, Classes, Code blocks and iterators, Pattern matching. Overview of Rails, Document requests, Processing forms, Rails applications with Databases, Layouts.

UNIT IV **10 H**

JDBC Overview – JDBC implementation – Connection class – Statements – Catching Database Results, handling database Queries. Networking– InetAddress class – URLclass- TCP sockets - UDP sockets, Java Beans –RM.

Java Servlets – life cycle of a servlet. The Servlet API, Handling HTTP Request and Response, using Cookies, Session Tracking. Introduction to JSP.

UNIT V **10 Hrs**

Introduction to Ajax: Overview of Ajax; The basics of Ajax; Rails with Ajax.

References:

1. 1 Thomas Powell, Web Design The complete Reference, Tata McGrawHill
2. Thomas Powell, HTML and XHTML The complete Reference, Tata McGrawHill
3. PHP for the Web: Visual Quick Start Guide, 4th Edition, Peachpit Press
4. Beginning PHP 5.3 (Wrox, free ebook: http://it-ebooks.info/book/713/)
5. P.J. Deitel and H.M. Deitel, Java for Programmers, Pearson education
6. Chris Bates, Web Programming Building Internet Applications, 3rd Edition, Wiley India,
7. Pragmatic Dave Thomas, Andy Thomas, et al., Programming Ruby: The Pragmatic Programmer's Guide,

126MCA03XXXCSC10T:Programming using Python

Teaching: 4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT I **10Hrs**

Installing Python, Simple program using Python, Expressions and Values, Variables and Computer Memory, error detection, Multiple line statements, Designing and using functions, functions provided by Python, Tracing function calls in memory model, omitting return statement. Working with Text: Creating Strings of Characters, Using Special Characters in Strings, Creating a Multiline String, Printing Information, Getting Information from the Keyboard.

UNIT II **10Hrs**

A Boolean Type , Choosing Statements to Execute, Nested If Statements , Remembering the Results of a Boolean Expression Evaluation , A Modular Approach to Program Organization, Importing Modules , Defining Your Own Modules, Testing Code Semi automatically Grouping Functions Using Methods: Modules, Classes, and Methods , Calling Methods the Object-Oriented Way, Exploring String Methods, Underscores.

UNIT III **12Hrs**

Storing Collections of Data Using Lists: Storing and Accessing Data in Lists, modifying Lists, Operations on Lists, Slicing Lists, Aliasing, List Methods, Working with a List of Lists. Repeating Code Using Loops: Processing Items in a List, Processing Characters in Strings, Looping Over a Range of Numbers, Processing Lists Using Indices, Nesting Loops in Loops, Looping Until a Condition Is Reached, Repetition Based on User Input, Controlling Loops UsingBreak and Continue Reading and Writing.

UNIT IV **10Hrs**

Files: Kinds of files, Opening a File, Techniques for Reading Files, Files over the Internet, Writing Files, and Writing Algorithms That Use the File-Reading Techniques, Multiline Records. Storing Data Using Other Collection Types: Storing Data Using Sets, Storing Data Using Tuples, Storing Data Using Dictionaries, Inverting a Dictionary, Using the In Operator on Tuples, Sets, and Dictionaries, Comparing Collections.

UNIT V **10Hrs**

Collection of New Information Object-Oriented Programming : Understanding a Problem Domain , Function “Isinstance,” Class Object, and Class Book , Writing a Method in Class Book, Plugging into Python Syntax: More Special Methods ,Creating Graphical User interface: Building a Basic GUI, Models, Views, and Controllers, Customizing the Visual Style Introducing few more Widgets, Object-Oriented GUIs, Keeping the Concepts from Being a GUI Mess.

Text Books:

1. Practical Programming: An introduction to Computer Science Using Python, second edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf.
2. Learning with Python: How to Think Like a Computer Scientist Paperback – Allen Downey , Jeffrey Elkner, 2015

Reference Books:

1. Introduction to Python for Computational Science and Engineering (A beginner's guide), Hans Fangohr.
2. Exploring Python, Timothy A. Budd, McGraw Hill Education
3. Python for Informatics: Exploring Information, Charles Severance.
4. Learning Python, Fourth Edition, Mark Lutz, O‟Reilly publication

126MCA03XXXSCC03T: Artificial Intelligence

Teaching: 4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT I **10 Hrs**

INTRODUCTION TO Al AND PRODUCTION SYSTEMS

Introduction to AI-Problem formulation, Problem Definition -Production systems, Control strategies, Search strategies. Problem characteristics, Production system characteristics - Specialized productions system- Problem solving methods – Problem graphs, Matching, Indexing and Heuristic functions -Hill Climbing-Depth first and Breath first, Constraints satisfaction – Related algorithms, Measure of performance and analysis of search algorithms.

UNIT II **10 Hrs**

REPRESENTATION OF KNOWLEDGE

Game playing – Knowledge representation, Knowledge representation using Predicate logic, Introduction to predicate calculus, Resolution, Use of predicate calculus, Knowledge representation using other logic-Structured representation of knowledge.

UNIT III **10Hrs**

KNOWLEDGE INFERENCE

Knowledge representation -Production based system, Frame based system. Inference – Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning – Certainty factors, Bayesian Theory-Bayesian Network-Dempster – Shafer theory.

UNIT IV **10 Hrs**

PLANNING AND MACHINE LEARNING

Basic plan generation systems – Strips -Advanced plan generation systems – K strips - Strategic explanations -Why, Why not and how explanations. Learning- Machine

UNIT V **12 Hrs**

EXPERT SYSTEMS

Expert systems – Architecture of expert systems, Roles of expert systems – Knowledge Acquisition – Meta knowledge, Heuristics. Typical expert systems – MYCIN, DART, XOON, Expert systems shells.

Text Books:

1. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, Mc Graw Hill- 2008.
2. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education, 2007.

Reference books

1. Peter Jackson, “Introduction to Expert Systems”, 3rd Edition, Pearson Education, 2007.
2. Stuart Russel and Peter Norvig “AI – A Modern Approach”, 2nd Edition, Pearson Education
3. Deepak Khemani “Artificial Intelligence”, Tata Mc Graw Hill Education 2013.

126MCA03XXXCSC11T: Cyber Forensic & Security

Teaching: 4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT 1: Understanding Cyber Forensics **12Hrs**

Computer forensics, Cyber forensics and Digital evidence, rules of evidence, Forensics analysis of e-mail- RFC282, Digital forensics life cycle, Chain of custody concept, Network forensics, Setting up a computer forensics laboratory, Computer forensics and steganography, Root kits, Information hiding, relevance of the OSI 7 layer model to computer forensics, Forensics and social networking sites: The security/privacy, Threats.

UNIT 2: Challenges in Cyber Forensics **12Hrs**

Technical challenges: understanding the raw data and its structure, The legal challenges in computer forensics and data privacy issues , Special tools and techniques - digital forensics tools, Special technique: data mining used in cyber forensics, Forensics auditing, Anti forensics.

UNIT 3: Security Principles and Practices **10Hrs**

Information system security principles, Threats and attacks, Classification of threats and assessing damages, Protecting information systems security, Information system security engineering process

UNIT 4: Security Threats **08Hrs**

Types of security threats- worms, viruses, Trojan horse, malware, malicious spyware, adware, botnet, spam, phishing, stack and buffer overflow

UNIT 5: Operating System Security **10Hrs**

Role of operating systems in information systems applications, Operating systems security, Patched operating systems, protected objects and methods of protection, Memory address protection, File protection mechanism

References:

1. Albert J. Marcella; Frederic Guillossou"**Cyber forensics: from data to digital evidence"** Hoboken, New Jersey: Wiley, 2012.

126MCA03XXXCSC12T:Internet of Things

Teaching:4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT I **10 Hrs**

Fundamentals of IoT: Introduction-Characteristics-Physical design - Protocols – Logical design – Enabling technologies – IoT Levels – Domain Specific IoTs – IoTvs M2M.

UNIT II **10 Hrs**

IoT Design Methodology: IoT systems management – IoT Design Methodology – Specifications Integration and Application Development.

UNIT III **12Hrs**

Building IoT With Raspberry PI: Physical device – Raspberry Pi Interfaces – Programming – APIs / Packages – Web services –

UNIT IV **10Hrs**

Building IoT with GALILEO/ARDUINO: Intel Galileo Gen2 with Arduino- Interfaces - Arduino IDE – Programming - APIs and Hacks

UNIT V **10Hrs**

Case Studies and Advanced Topics: Various Real time applications of IoT- Connecting IoT to cloud – Cloud Storage for IoT – Data Analytics for IoT – Software & Management Tools for IoT

References:

1. ArshdeepBahga, Vijay Madisetti, “Internet of Things – A hands-on approach”, Universities Press, 2015.
2. Manoel Carlos Ramon, “Intel® Galileo and Intel® Galileo Gen 2: API Features and Arduino Projects for
Linux Programmers”, Apress, 2014. 3. Marco Schwartz, “Internet of Things with the Arduino Yun”, Packt Publishing, 2014

126MCA03XXXCSC13T:Pattern Recognition

Teaching:4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT I 08Hrs

Basics of Probability, Random Processes and Linear Algebra (recap): Probability: independence of events, conditional and joint probability, Bayes theorem Random Processes: Stationary and non-stationary processes, Expectation, Autocorrelation, Cross-Correlation, spectra.

UNIT II 10Hrs

Linear Algebra: Inner product, outer product, inverses, eigen values, eigen vectors, singular values, singular vectors. Bayes Decision Theory : Minimum-error-rate classification. Classifiers, Discriminant functions, Decision surfaces. Normal density and discriminant functions. Discrete features.

UNIT III 12Hrs

Parameter Estimation Methods : Maximum-Likelihood estimation :Gaussian case. Maximum a Posteriori estimation. Bayesian estimation: Gaussian case. Unsupervised learning and clustering - Criterion functions for clustering. Algorithms for clustering: K-Means, Hierarchical and other methods. Cluster validation. Gaussian mixture models, Expectation-Maximization method for parameter estimation. Maximum entropy estimation. Sequential Pattern Recognition. Hidden Markov Models (HMMs). Discrete HMMs. Continuous HMMs. Nonparametric techniques for density estimation. Parzen-window method. K-Nearest Neighbour method.

UNIT IV 10Hrs

Dimensionality reduction: Principal component analysis - it relationship to eigen analysis. Fisher discriminant analysis - Generalised eigen analysis. Eigen vectors/Singular vectors as dictionaries. Factor Analysis, Total variability space - a dictionary learning methods. Non negative matrix factorisation - a dictionary learning method..

UNIT V 10Hrs

Linear discriminant functions : Gradient descent procedures, Perceptron, Support vector machines - a brief introduction. Artificial neural networks: Multilayer perceptron - feedforwark neural network. A brief introduction to deep neural networks, convolutional neural networks, recurrent neural networks.

Text Book:

1. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001
2. S.Theodoridis and K.Koutroumbas, Pattern Recognition, 4th Ed., Academic Press, 2009
3. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006

126MCA03XXXCSC14T :Embedded Systems

Teaching:4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

Unit-I **10hrs**

Embedded System: Embedded vs General computing system, classification, application and purpose of ES. Core of an Embedded System, Memory, Sensors, Actuators, LED, Opto coupler, Communication Interface, Reset circuits, RTC, WDT, Characteristics and Quality Attributes of Embedded Systems.

Unit-II **12hrs**

Hardware Software Co-Design, embedded firmware design approaches, computational models, embedded firmware development languages, Integration and testing of Embedded Hardware and firmware, Components in embedded system development environment (IDE), Files generated during compilation, simulators, emulators and debugging

Unit-III **10hrs**

ARM-32 bit Microcontroller: Thumb-2 technology and applications of ARM, Architecture of ARM Cortex M3, Various Units in the architecture, General Purpose Registers, Special Registers, exceptions, interrupts, stack operation, reset sequence.

Unit-IV **10hrs**

Instruction Sets: Assembly basics, Instruction list and description, useful instructions, Memory Systems, Memory maps, Cortex M3 implementation overview, pipeline and bus interface.

Unit-V **10hrs**

Exceptions, Nested Vector interrupt controller design, Systick Timer, Cortex-M3 Programming using assembly and C language, CMSIS.

Text Books:

1. K. V. Shibu, "Introduction to embedded systems", TMH education Pvt. Ltd. 2009.
2. Joseph Yiu, ―The Definitive Guide to the ARM Cortex-M3‖, 2ndedn, Newnes, (Elsevier), 2010.

Reference Book:

James K. Peckol, "Embedded systems- A contemporary design tool", John Wiley, 2008.

126MCA03XXXCSC15T:Cloud Computing

Teaching:4hrs./week **Max. Marks:80**
Credits: 04Hrs.:52 **I. A. Marks:20**

UNIT I **10 Hrs**

Cloud Computing Basics: Cloud Computing Overview, Applications, Intranets and the Cloud, First Movers in the cloud.

UNIT II **10 Hrs**

Organization and Cloud Computing with the Titans: When You Can Use Cloud Computing, Benefits, Limitations, Security Concerns, Regulatory Issues. – Google, EMC, NetApp, Microsoft, Amazon, Salesforce.com, IBM, Partnerships.

The Business Case for Going to the Cloud: Cloud Computing Services, How Those Applications Help Your Business, Deleting Your Datacenter, Salesforce.com, Thomson Reuters.

UNIT III **10 Hrs**

Hardware and Infrastructure: Clients, Security, Network, Services.

Accessing the Cloud: Applications, Web APIs, Web Browsers.

Cloud Storage: Overview, Cloud Storage Providers, Standards – Applications, Client, Infrastructure, Service, software.

UNIT IV **12 Hrs**

Software as a Services: Overview, Driving Forces, Company Offerings, Industries.

Software plus Services: Overview, Mobile Device Integration, Providers, Microsoft Online.

Developing Applications: Google, Microsoft, Intuit Quick Base, Cast Iron Cloud, Bungee Connect, Development, Troubleshooting, Application Management.

UNIT V **10 Hrs**

Local Clouds and Thin Clients: Virtualization in Your Organization, Server Solutions, Tin Clients, Cast Study: McNeilus Steel.

Migrating to the Cloud: Cloud Services for Individuals, Cloud Services Aimed at the Mid-Market, Enterprise-Class Cloud Offerings, Migration, Best practices and the future of cloud computing.

Text Books:

1. Anthony T. Vete, Toby J. Velte, Robert Elsenpeter, ―Cloud Computing A Practical Approach‖, McGraw-Hill, 2010.

References:

1. Barrie Sosinsky, ‖Cloud computing Bible‖, Wiley Publications, 1st Edition, 2011.
2. A. John Rhoton, ―Cloud computing explained‖, Recursive press, 2010.

Syllabus of IV Semester MCA programme,
CHOICE BASED CREDIT SYSTEM (CBCS)
(According new regulations w.e.f. 2024-25)

IV SEMSTER MCA w.e.f. 2024-25								
Course	Subject Name	Teaching Hrs per Week	Practical Hrs/week	Examination				Credits
				Duration (Hrs.)	Marks			
					Theory/ Practical	IA	Total	
126MCA04XX XCSC01P	Project/ Technical Seminar	-	--	--	--	100	100	2
126MCA04XX XCSC02P	Project (16 Weeks)	-	--	3	300	100	400	20
Total							**500**	**22**

TS: Technical Seminar PJ: Project

Project / Technical Seminar:

I A marks shall be awarded by a research committee comprising of Chairman of the Department, Guide/co-guide, if any, and a senior faculty of the department. Participation in the seminar by all postgraduate students of the programme shall be mandatory. The marks awarded for Technical Seminar, shall be based on the evaluation of Seminar Report, Presentation skills and performance in Question and Answer session in the ratio 50:25:25.

Students may be assigned to do literature survey of existing work on contemporary topics and present in front of the research committee (compulsory). Student shall highlight on the research gap and propose solution.

Project:

The candidate should carry out the project in any industry or R&D institution or educational institution under a guide/co-guide. The candidate has to present the work carried out before the examiners during the University examination. The work carried out should be free from plagiarism. The literature study may be clearly written which may be summary of existing project and highlight of what are the functionalities that are proposed to this project. Student shall indicate the different research papers, documents refereed as a part of the literature study. This is an individual project for a duration of minimum of 4 months or duration of the semester. Paper publication in an indexed journal/conference is compulsory as part of the project work.