TEMPLATE FOR COURSE SYLLABUS FOR NEP IMPLEMENTATION

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: UCMSMAJ24006 (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III ☐ Semester IV [√]

Semester V ☐ Semester VI ☐ Semester VII ☐ Semester VIII ☐

Course Name: DATA BASE MANAGEMENT SYSTEM (DBMS)

Course Code:

Course Credit: (Will be provided by the University)

Marks Allotted:

Theoretical	3	Practical/Tutorial	1
Theoretical	40	Practical/Tutorial	20
Continuing Evaluation	10	Attendance	5

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity? YES [√] NO ☐

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: UCMSMAJ24006

Course Name: DATA BASE MANAGEMENT SYSTEM

Brief Course Description:

This course provides a comprehensive introduction to the principles, design, and implementation of Database Management Systems (DBMS). Students will learn the fundamental concepts of databaseorganization, data modeling, and SQL (Structured Query Language). The course emphasizes both theoretical understanding and practical skills necessary for effective database design, implementation, and management.

Prerequisite(s) and/or Note(s):

Basic understanding of computer systems, data structures, and programming concepts.

Course Objectives:

Knowledge acquired:

Upon completion of the course, students should be able to design, implement, and manage relational databases effectively. They will gain practical skills in SQL, understand database optimization techniques, and be prepared for roles in database administration, database development, and related fields.

Skills gained:

Studying Database Management Systems (DBMS) provides individuals with a variety of skills that are valuable in the field of information technology, software development, and data management. Here are key skills gained from learning DBMS: Database design, SQL proficiency, data modeling, query optimization, Normalization Database security etc.

These skills collectively prepare individuals for roles in database administration, database development, data analysis, and other areas where a deep understanding of DBMS is crucial for effective data management and utilization.

Competency Developed:

Studying Database Management Systems (DBMS) develops a range of competencies that are essential for individuals pursuing careers in information technology, database administration, software development, and data management. Here are key competencies developed from learning DBMS: Database connectivity, Backup and Recovery, Database administration, Data warehousing, Data Mining, Distributed database etc. These competencies collectively prepare individuals for roles as database administrators, database developers, data analysts, and other positions where a deep understanding of DBMS is fundamental for effective data management and utilization.

Detailed Syllabus

2nd Year: Semester 4

UCMSMAJ24006: DATABASE MANAGEMENT SYSTEM [Credits: 3, Lectures 45]

Unit 1: Introduction (10 Lectures)
Definition of DBMS, file processing system Vs DBMS, Limitation of file processing system, database users, Characteristics of database, database components, database system architecture- 3-layer and 2- layers, data independence, data dictionary. View of Data – Data Abstraction – Instances and Schemas diagrams. DBA- Database Administrator- roles and responsibilities of a DBA

Unit 2: Database models (7 Lectures)
A brief overview of three models- Hierarchical model, Network model and relational model, comparison of three models.

Unit 3: **Database Design, ER-Diagram (12 Lectures)**
ER-Model, Constraints, ER-Diagrams, weak entity sets, E. F. Codd's rules, Keys-Primary, Candidate, Composite, Alternate, Foreign, Super, Relational Schemas, Normalization: Lossless decomposition, lossy decomposition, functional dependencies, Normal Forms- (Up to BCNF -1NF, 2NF, 3NF, BCNF).

Unit 4: Transaction Processing (4 Lectures)
ACID properties, concurrency control.

Unit 5: SQL and ORACEL (12 Lectures)
Oracle commands, Oracle datatypes, operators- Arithmetic, Logical, Range searching, Pattern Matching, Oracle functions- Aggregate, Numeric, String, Conversion, Date and Time, Oracle table- Dual, Grouping data-Concept of grouping, Group by Clause, Having clause Language-DDL, DML, DCL, SQL Functions- queries and sub-queries in SQL, Constrains and indexes in SQL.

Suggested Readings:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education,2013.
5.Ivan Bayross, Teach Yourself SQL & PL/SQL Using ORACLE 8i & 9i with SQLJ,1st Edition, BPB Publications,2003

UCMSMAJ24006: DATABASE MANAGEMENT SYSTEM LAB [Credits: 1, Lab Hours: 30]

Students are advice to do laboratory/practical practice not limited to, but including the following types of problems.

1. Demonstrate table creation
2. INSERT data into the table
3. SELECT the record from the table
4. SELECT all rows and all columns form the table
5. Demonstrate filtering table data – Selected columns and all rows, selected rows and all columns, selected columns and selected rows, Rename table.
6. SELECT all records from the table eliminating duplicate rows.
7. Filtering table data
8. Sorting data in a table
9. Creating a table from a another table
10. Inserting data into a table from another table.
11. Demonstrate delete operations
12. Updating the contents of a table
13. Modifying the structure of table
14. Destroying tables
15. Find out the table created by different users
16. Demonstrate different types of data constraints
17. Demonstrate different operators in SQL-Arithmetic, logical,
18. Demonstrate pattern matching, range searching
19. Demonstrate oracle table 'DUAL'
20. Demonstrate different oracle functions.

TEMPLATE FOR COURSE SYLLABUS FOR NEP IMPLEMENTATION

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: UCMSMAJ24007 (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III ☐ Semester IV [√]

Semester V ☐ Semester VI ☐ Semester VII ☐ Semester VIII ☐

Course Name: SOFTWARE ENGINEERING

Course Code:

Course Credit: (Will be provided by the University)

Marks Allotted:

Theoretical	3	Practical/Tutorial	1
Theoretical	60	Practical/Tutorial	0
Continuing Evaluation	10	Attendance	5

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity? YES ☐ NO [√]

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: UCMSMAJ24007

Course Name: SOFTWARE ENGINEERING

Brief Course Description:

This course provides a comprehensive introduction to the principles, methodologies, and practices of software engineering. Students will learn the entire software development lifecycle, from requirements gathering to design, implementation, testing, deployment, and maintenance. Emphasis is placed on adopting systematic and disciplined approaches to software development to produce high-quality and reliable software systems.

Prerequisite(s) and/or Note(s):

Basic understanding of programming and software concepts.

Course Objectives:

Knowledge acquired:

Upon completion of the course, students should be able to understand and apply software engineering principles, effectively manage the software development lifecycle, and contribute to the production of high-quality software systems. The course prepares students for roles in software development, project management, and related fields.

Studying software engineering provides individuals with a diverse set of knowledge and skills that are crucial for designing, developing, and maintaining high-quality software systems. Here are key knowledge areas gained from software engineering: SDLC, System Design, Testing, software configuration management etc. These knowledge areas collectively equip individuals with a holistic understanding of the software development process and prepare them for roles in software engineering, project management, and related fields.

Skills gained:

Studying software engineering provides individuals with a diverse set of knowledge and skills that are crucial for designing, developing, and maintaining high-quality software systems. Here are key knowledge areas gained from software engineering: SDLC, System Design, Testing, software configuration management etc. These knowledge areas collectively equip individuals with a holistic understanding of the software development process and prepare them for roles in software engineering, project management, and related fields.

Competency Developed:

Studying software engineering equips individuals with a broad set of skills that are essential for designing, developing, and maintaining software systems. Here are key skills gained from software engineering: Programming skill, problem solving skill, algorithmic thinking, software development etc. These skills collectively contribute to a well-rounded foundation in software engineering, enabling individuals to contribute effectively to the development of robust and high-quality software systems.

Detailed Syllabus

2nd Year: Semester 4

UCMSMAJ24007: SOFTWARE ENGINEERING [Credits: 3, Lectures 45]

Unit 1: Introduction (4 Lectures)
Definition of software and software engineering, the Evolving Role of Software, Software Characteristics, Software quality, Changing Nature of Software, Software process.

Unit 2: Software development models (6 Lectures)
Software Development Life Cycle (SDLC), Types of Software Models- Waterfall Model, Iterative Model, Spiral Model, Prototype Model. Capability Maturity Model Integration (CMMI).

Unit 3: Software Requirement Analysis and Planning (10 Lectures)
Software Requirement Analysis and Modeling Techniques, Software Requirement Specification (SRS)-Need for SRS, Characteristics and Components of SRS, Fact finding techniques, Cost Estimation-COCOMO Model, Data Flow Diagram (DFD) and Entity-Relationship Diagram (ERD),

Unit 4: Software Project Management -SPM (8 Lectures)
Define SPM, Needs of SPM, Software Risks, Risk Identification, RMMM Plan, Software Quality Control and Quality Assurance, ISO (International Standards Organization) 9000 Certification, Software Metrics.

Unit 5: Coding (4 Lectures)
Define Coding, Goals of Coding, characteristics of Programming Language, Programming style, Coding Standards, Coding Guidelines.

Unit 6: Testing Strategies & Tactics (13 Lectures)
Software Testing Fundamentals-Test Plan, Test Execution, Test Review, Inspection and Auditing, Testing principles, software Verification and validation, Types of software testing - Black-Box Testing, White-Box Testing and their type, Other testing- Regression Testing, Mutation Testing, Acceptance testing, Alpha testing, Beta testing.

Suggested Readings:

1. R.S. Pressman, Software Engineering: A Practitioner's Approach (7th Edition), McGraw-Hill, 2009.
2. P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa Publishing House, 2003.
3. K. K. Aggarwal and Y. Singh, Software Engineering (2nd Edition), New Age International Publishers,2008.
4. I. Sommer Ville, Software Engineering (8th edition), Addison Wesley, 2006.
5. D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005.
6. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India, 2004.

UCMSMAJ24007: SOFTWARE ENGINEERING TUTORIAL [Credits: 1, Tutorial Hours: 30]

Software Engineering tutorial as assigned and advised by teacher(s)

TEMPLATE FOR COURSE SYLLABUS FOR NEP IMPLEMENTATION

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: UCMSMAJ24008 (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III ☐ Semester IV [√]

Semester V ☐ Semester VI ☐ Semester VII ☐ Semester VIII ☐

Course Name: PROGRAMMING IN JAVA

Course Code:

Course Credit: (Will be provided by the University)

Marks Allotted: Theoretical 3 Practical/Tutorial 1

Theoretical 40 Practical/Tutorial 20

Continuing Evaluation 10 Attendance 5

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity ? YES [√] NO ☐

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: UCMSMAJ24008

Course Name: PROGRAMMING IN JAVA

Brief Course Description:

This course serves as an introduction to programming using the Java programming language. Students will learn fundamental concepts of software development, object-oriented programming (OOP), and Java syntax. The course covers essential topics such as variables, data types, control structures, classes, objects, inheritance, polymorphism, and exception handling. Practical exercises and projects reinforce theoretical knowledge and help students develop the skills necessary for Java application development.

Prerequisite(s) and/or Note(s):

Basic understanding of programming concepts is recommended but not mandatory.

Course Objectives:

Knowledge acquired:

Upon completion of the course, students should be able to write Java programs, understand and apply object-oriented principles, design and implement simple graphical user interfaces, work with files and databases using Java, and have a foundational understanding of Java web development concepts. This course prepares students for further studies in Java development and related fields.

Skills gained:

Studying Java programming imparts a variety of skills that are valuable in software development and related fields. Here are key skills gained from learning and practicing Java programming: OOP, Data structure, Exception handling, Multithreading, File I/O Operation, development, Networking etc. These skills collectively prepare individuals for roles in software development, web development, mobile app development, and other areas where Java is widely used.

Competency Developed:

Studying Java programming leads to the development of various competencies that are essential for a career in software development and related fields. Here are key competencies developed from learning and practicing Java: OOP, Data structure implementation, GUI development, Database connectivity, version control etc. These competencies collectively prepare individuals for roles as Java developers, software engineers, and contributors to various aspects of the software development process.

UCMSMAJ24005: Programming in Java [Credits:3, Lectures:45]

Unit 1: Introduction to Java (7 Lectures)
Java Architecture and Features, Understanding the semantic and syntax differences between C and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords, Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, JavaMethods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Checking, Built-in Java Class Methods).

Unit 2: Arrays, Strings, and I/O (5 Lectures)
Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, ManipulatingStrings, String Immutability& Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

Unit 3: Object-Oriented Programming Overview (5 Lectures)
Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

Unit 4: Inheritance, Interfaces, Packages, Enumerations, Autoboxing & Metadata (11 Lectures)
Inheritance:(Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, extending interfaces and packages, Package and Class Visibility, Using Standard JavaPackages (util, lang, io, net), Wrapper Classes, Autoboxing /Unboxing, Enumerations and Metadata.

Unit 5: Exception Handling and Threading (10 Lectures)
Exception types, uncaught exceptions, throw, built-in exceptions, creating your own exceptions;Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads.

Unit 6: Applets and Event Handling (7 Lectures)
Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. The design and implementation of GUIs using the AWTcontrols, Swing components of Java Foundation Classes such as labels, buttons, text fields, layoutmanagers, menus, events, and listeners; Graphic objects for drawing figures such as lines,rectangles, ovals, using different fonts.

Suggested Readings:

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4thEdition, 2005.
2. James Gosling, Bill Joy, GuyL Steele Jr, Gilad Bracha, Alex Buckley "The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
3. Joshua Bloch, "Effective Java" 2nd Edition, Publisher: Addison-Wesley,2008.
4. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume1, 9thEdition, PrinticeHall.2012
5. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features", 9th Edition, PrinticeHall.2013
6. Bruce Eckel, Thinking in Java", 3rd Edition, PHI,2002.
7. E. Balaguruswamy,"Programming with Java", 4th Edition, McGrawHill.2009.
8. Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall,2011.
9. Bert Bates, Kathy Sierra, "Head First Java", Orielly Media Inc. 2nd Edition,2005.
10. Object Oriented Programming through JAVA, Pradha Krishna, University Press
11. David J. Eck, "Introduction to Programming Using Java", Published by Create Space Independent Publishing Platform, 2009.
12. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2ndEdition, 2004.
13. R.Nageswara,CoreJava: An Integrated Approach, Dreamtech Press

UCMSMAJ24005: Programming in Java [Credit: 1, Lab Hours: 30]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Java Program to Get Input from the User
2. Java Program to Swap Two Numbers
3. Java Program to Find the Factorial of a Number
4. Java Program to Check if a number is even or odd
5. Java Program to Find Largest Among 3 Numbers
6. Java Program to Find LCM and HCF of 2 numbers
7. Java Program to Display All Prime Numbers from 1 to N
8. Java Program to Check Leap Year
9. Java Program to Check Armstrong Number between Two Integers
10. Java Program to Check whether input character is vowel or consonant
11. Java Program to Find Factorial of a number
12. Java Program to Display Fibonacci Series
13. Java Program to Calculate Simple Interest
14. Java Program to Print Pattern
15. Java Program for Conversions
16. Java Program to Create a Class and Object
17. Java Program to Create Abstract Class
18. Java Program to Show Inheritance in Class
19. Java Program to Show Overloading of Methods in Class
20. Java Program to Show Overriding of Methods in Classes
21. Java Program to Show Use of Super Keyword in Class
22. Java Program to Show Use of This Keyword in Class
23. Java Program to Show Usage of Static keyword in Class
24. Java Program to Show Use of Static and Non-static Methods
25. Java Program to Show Usage of toString() Method
26. Java Program to Show Usage of compare() Method
27. Java Program to Show Usage of equals() Method
28. Java Program to implement a thread.
29. Java Program to Search an Element in an Array
30. Java Program to Find the Largest and smallest Element in an Array
31. Java Program to Sort an Array
32. Java Program to Merge Two Arrays
33. Java Program to Add, subtract, multiply Two Matrices
34. Java Program to Check Whether Two Matrices Are Equal or Not
35. Java Program to Find the Transpose
36. Java Program to Compute the Sum of Diagonals of a Matrix
37. Java Program to find the sum of upper and lower triangular
38. Java Program to Get a Character From the Given String
39. Java Program to Replace a Character at a Specific Index
40. Java Program to Reverse a String
41. Java Program to Compare two strings
42. Java Program to implement other string functions
43. Java Program to find the sum of any number of integers entered as command line arguments
44. Java Program to find the sum of any number of integers interactively, i.e., entering every number from the keyboard, where as the total number of integers is given as a command line argument
45. Java Applet programs