VIDYASAGAR UNIVERSITY

Midnapore, West Bengal

PROPOSED CURRICULUM & SYLLABUS (DRAFT) OF

BACHELOR OF SCIENCE WITH COMPUTER SC.

(MULTIDISCIPLINARY STUDIES)

3-YEAR UNDERGRADUATE PROGRAMME

(w.e.f. Academic Year 2023-2024)

Based on

Curriculum & Credit Framework for Undergraduate Programmes

(CCFUP), 2023 & NEP, 2020

VIDYASAGAR UNIVERSITY

BACHELOR OF SCIENCE IN MULTIDISCIPLINARY STUDIES with COMPUTER SCIENCE

(*Under CCFUP, 2023*)

Level	YR.	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
								CA	ESE	TOTAL
B.Sc. in Physical Sc./ Math. & Comp. Sc. with Computer Science	2nd	III	**SEMESTER-III**							
			Major-A2	COSPMJ02	**T: Data Structure & Algorithm;** *(To be studied by students taken* **Computer Sc.** *as Discipline- A)*	4	3-1-0	15	60	75
			Major-A3	COSPMJ03	**T: OOPS Using C++** *(To be studied by students taken* **Computer Sc.** *as Discipline- A)*	4	3-1-0	15	60	75
			SEC	SEC03	*To be taken from SEC-03 of Discipline C.*	3	0-0-3	10	40	50
			AEC	AEC03	Communicative English-2 (***common for all programmes***)	2	2-0-0	10	40	50
			MDC	MDC03	Multidisciplinary Course-3 (***to be chosen from the list***)	3	3-0-0	10	40	50
			Minor-3 (Disc.-C3)	COSMIN03	**T: Digital Logic;** *(To be studied by students taken* **Computer Sc.** *as Discipline- C)*	4	3-1-0	15	60	75
					Semester-III Total	20				375
		IV	**SEMESTER-IV**							
			Major-B2		*To be decided (Same as MajorA2 for* **Computer Sc.** *taken as Discipline-B)*	4	3-1-0	15	60	75
			Major-B3		*To be decided (Same as Major–A3 for* **Computer Sc.** *taken as Discipline-B)*	4	3-1-0	15	60	75
			Major (Elective) -1	COSMJE-01	**T: Python** *(To be studied by students taken* **Computer Sc.** *as Discipline- A)*	4	3-1-0	15	60	75
			AEC	AEC04	MIL-2 (***common for all programmes***)	2	2-0-0	10	40	50
			Minor -4 (Disc.-C4)	COSMIN04	**T: Data Structure; P: Practical** *(To be studied by students taken* **Computer Sc.** *as Discipline- C)*	4	3-0-1	15	60	75
			Summer Intern.	IA	Internship / Apprenticeship- **activities to be decided by the Colleges following the guidelines to be given later**	4	0-0-4	-	-	50
					Semester-IV Total	22				400
					TOTAL of YEAR-2	42	-	-	-	775

MJP = Major Programme (Multidisciplinary), MI = Minor, A/B = Choice of Major Discipline; C= Choice of Minor Discipline; SEC = Skill Enhancement Course, AEC = Ability Enhancement Course, MDC = Multidisciplinary Course, CA= Continuous Assessment, ESE= End Semester Examination, T = Theory, P= Practical, L-T-P = Lecture-Tutorial-Practical, MIL = Modern Indian Language

MAJOR (MJ)

MJ A2/B2: Data Structure & Algorithm **Credits 04 (FM: 75)**

OBJECTIVE OF THE COURSE **(Theory: 45 Lectures)**

- Introduce fundamental concepts and importance of data structures in computing.
- Learn to implement linear data structures such as arrays, linked lists, stacks, and queues.
- Explore non-linear data structures including trees, graphs, and heaps.
- Emphasize analyzing algorithm efficiency in terms of time and space complexity.
- Develop skills in designing and implementing efficient algorithms.
- Provide hands-on programming experience with languages like C or Python.

OUTCOME OF THE COURSE

- Students will be able to implement and utilize fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
- Students will be able to design, analyze, and optimize algorithms for a variety of problems, considering both time and space complexity to ensure efficient solutions.
- Students will be able to evaluate and compare the efficiency of algorithms..
- Students will gain the capability to implement and apply advanced data structures, such as heaps, hash tables, and trees, to solve complex problems.
- Students will acquire hands-on experience in coding data structures and algorithms in a programming language.

MJ A2/B2T: Data Structure & Algorithm **Credits 03 (45 Lectures)**

Course contents:

Module- I: Array **02 Hrs.**

Arrays: Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation), Row major and column major operation (2D).

Module- II: Basic Data Structures **10 Hrs.**

Stacks: Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack. Queues: Array and Linked representation of Queue, De-queue, Priority Queues.

Module- III: Linked List **04 Hrs.**

Linked Lists: Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists.

Module- IV: Recursion **03 Hrs.**

Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

Module- V: Tree and Graph **10 Hrs.**

Trees: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals in Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees), Tree traversal techniques, Heap Sort. Graph: Introduction to Graphs, Breadth first search and connected components, Depth first search in directed and undirected graphs and strongly connected components

Module- VI: Hashing **02 Hrs.**
Hashing: Hashing technique, Collision resolution, chaining Different types of hash function.

Module-VII: Introduction to Algorithm **02 Hrs.**
Asymptotic Notations, Time complexity, Space complexity, Algorithm design techniques.

Module- VIII: Searching and Sorting Algorithms **08 Hrs.**
Searching and Sorting: Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Bubble Sort, Quick Sort and Merge sort, Comparison of Sorting Techniques. Heap: Heapsort.

Module- IX: Spanning Tree Algorithms **04 Hrs.**
Spanning trees: Prim's and Kruskal's algorithm, union-find data structure. Dijkstra's algorithm for shortest path. shortest path tree. Shortest and longest paths in directed acyclic graphs.

Suggested Readings:

1. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (CLRS)
2. Sartaj Sahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
3. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using C and C++:, Second edition, PHI, 2009.
4. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
5. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley, 2013
6. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub, 2003
7. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 edition, 2009.

MJ A3/B3: OOPS Using C++ **Credits 04 (FM: 75)**

OBJECTIVE OF THE COURSE

- Provide a deep understanding of object-oriented programming principles and their application using C++.
- Cover fundamental concepts such as classes, objects, inheritance, polymorphism, and encapsulation.
- Teach students to design and implement reusable and modular code using OOP principles.
- Emphasize the creation and manipulation of complex data structures through dynamic memory management and operator overloading.
- Develop practical programming skills through hands-on projects and assignments involving real-world applications.
- Introduce advanced C++ features such as templates and the Standard Template Library (STL) for efficient coding practices.
- Provide experience in debugging and testing C++ applications to ensure reliability and performance.
- Equip students with the skills to develop robust, efficient, and maintainable software using OOP techniques in C++.
- Prepare students for advanced programming and software development roles.

MJ A3/B3T: OOPS Using C++ **Credits 04**

Course contents:

MODULE - I

Concepts of OOP: Introduction OOP, Procedural Vs Object Oriented Programming, Principles of OOP, Benefits and applications of OOPS

MODULE - II

C++ Basics - Objects and Classes: Basics of object and class in C++, Private and public members, static data and function members, constructors and their types, destructors, operator overloading, type conversion

MODULE - III

Inheritance: Concept of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members, overriding, virtual base class- Polymorphism - Pointers in C++, Pointes and Objects, this pointer, virtual and pure virtual functions, implementing polymorphism - Introduction to exception, try-catch-throw, multiple catch, catch all, re-throwing exception, implementing user defined exceptions

Suggested Readings:

1. E Balagurusamy , Object Oriented Programming with C++, 5 th edition, Tata McGraw, 2011.
2. Deitel and Deitel , “C++: How to Program”, 9th Edition, Pearson, 2013.

Major Elective (MJE)-01: PYTHON **Credits 04 (FM: 75)**

OBJECTIVE OF THE COURSE

The objectives of this course are to make the student understand programming language, programming, concepts of Loops, reading a set of Data, stepwise refinement, Functions, Control structure, Arrays. After completion of this course the student is expected to analyze the real-life problem and write a program in 'Python' language to solve the problem. The main emphasis of the course will be on problem solving aspect i.e., developing proper algorithms.

- After completion of the course the student will be able to
- Develop efficient algorithms for solving a problem.
- Use the various constructs of a programming language viz. conditional, iteration and recursion.
- Implement the algorithms in "Python" language.
- Use simple data structures like arrays, stacks andlist in solving problems.

OUTCOME OF THE COURSE

- Master basic Python syntax, including variables, data types, and control structures.
- Work with Python's built-in data structures like lists, tuples, dictionaries, and sets.
- Write reusable code using functions and organize code using modules and packages.
- Perform file operations such as reading from and writing to files.
- Implement error and exception handling to write robust code.
- Understand and apply object-oriented programming (OOP) principles like classes, inheritance, and polymorphism.
- Use popular Python libraries like NumPy, Pandas, and Matplotlib for data analysis and visualization.
- Automate repetitive tasks using Python scripts.
- Explore Python's extensive libraries and frameworks for web development, data science, machine learning, etc.
- Use Git and GitHub for version control and collaborative development.
- Develop skills to test, debug, and optimize Python code.
- Apply all learned concepts in a real-world project to solve a practical problem using Python.

MJE-01T: PYTHON **Credits 04**

Course Contents:

Planning the Computer Program **2 Hrs.**

Concept of problem solving, Problem definition, Problem design, Debugging, Types of Errors in programing, Documentation

Techniques of Problem Solving **2 Hrs.**

Flowcharting, decision table, algorithms, Structured programing concepts, Programing methodologies viz. top-down and bottom-up

Overview to Python Programming **2 Hrs.**

Structure of Python Program, Elements of Python

Introduction to Python **4 Hrs.**

Python Interpreter, Python shell, Indentation, Atoms, Identifiers and keywords, literals, Strings, Operator (Arithmetic Operator, Relational Operator, Logical or Boolean Operator, Assignment Operator, Ternary operator, Bitwise Operator)

Creating Python Programs: **20 Hrs.**

Input and Output Statements, Control Statements (Branching, Looping, Conditional Statement, Exit, Function, Difference, between break, continue and pass). Defining Functions, Default arguments and Exception handling

Iterations and Recursions **10 Hrs.**

Conditional execution, Alternative execution, Nested conditionals, Return statements, Recursion, Stack diagrams for recursive functions, Multiple assignment, While statement, For statement.

String and List **15 Hrs.**

String as a compound data type, Length, Traversal and the for loop, String slices, String Comparison, A find function, Looping and counting, List values, Accessing elements, List length, List membership, List and for loops, List operations, List deletion, Cloning lists, Nested Lists

Object Oriented Programing **05 Hrs.**

Introduction to Classes, Objects and Methods, Standard Libraries

Suggested Readings:

1. Jhon V. Guttag, "Introduction to Computation and Programming Using Python", MIT Press
2. Allen Downey, "Think Python: How to Think a Computer Scientist", O'Reilly
3. Mark Lutz, "Learning Python, 5th Edition", O'Reilly

MINOR (MI)

MI-3/C3: Same as Minor-3 (COSMIN03) of Computer Science (Hons) programme **Credits 04**
Full Marks: 75

MI-4/C4: Same as Minor-2 (COSMIN04) of Computer Science (Hons) programme **Credits 04**
Full Marks: 75

SKILL ENHANCEMENT COURSE (SEC)

(To be studied by students taken Computer Science as Discipline- C)

SEC-03 P: Same as SEC-03 (COSSEC03) of Computer Science (Hons) programme **Credits 03**
Full Marks: 50