



KOLHAN UNIVERSITY

Chaibasa, Jharkhand, India

Proposed Syllabus for
Four Year Undergraduate Programme (FYUGP)
of

***Bachelor of Computer Application
(BCA)***

Semester - 2

With Effect From
Academic Year 2022 - 2023

As Per Revised Curriculum and Credit Framework for the
FYUGP under the provisions of NEP - 2020

Table of Contents

Serial Number	Topics	Page Number
1	Course Structure (Semester – II) for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)	03 – 03
2	MN–2A (Theory): Entrepreneurship Development	04 – 05
3	MN–2A (Practical): Entrepreneurship Development Lab	06 – 06
4	MJ–2 (Theory): Object Oriented Programming with C++	07 – 08
5	MJ–3 (Theory): Data Structure using C++	09 – 10
6	MJ (Practical-2): Data Structure and C++ Lab	11 – 14

**Course Structure (Semester – II) for Four Year Undergraduate Programme (FYUGP)
of
Bachelor of Computer Applications (BCA)**

Sem.	Paper Code	Paper Title	L–T–P	Credits	Contact Hours
II	AEC–2	<i>Language and Communication Skills: English Composition</i>		2	
	SEC–2	<i>Skill Enhancement Course 2: Mathematical & Computational Thinking Analysis</i>		3	
	MDC–2	<i>Multi-Disciplinary Course – 2</i> <i>(To be selected by the students from the list of available options)</i>		3	
	MN–2A (Theory)	Entrepreneurship Development	3–0–0	3	45
	MN–2A (Practical)	Entrepreneurship Development Lab	0–0–1	1	30
	MJ–2 (Theory)	Object Oriented Programming with C++	3–0–0	3	45
	MJ–3 (Theory)	Data Structure using C++	3–0–0	3	45
	MJ (Practical-2)	Data Structure and C++ Lab	0–0–2	2	60
Total Credits				20	

Abbreviations:

L–T–P (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC** (Value Added Course), **SEC** (Skill Enhancement Course), **MDC** (Multi Disciplinary Course), **MN–1** (Minor From Discipline–1), **MN–2** (Minor From Vocational Studies/Discipline–2), **IAP** (Internship/Apprenticeship/Project), **MJ** (Major Disciplinary/Interdisciplinary Courses)

MN-2A (Theory): Entrepreneurship Development

3 Credits | 45 Minimum Class Hours | Semester II

Objectives:

It provides exposure to the students to the entrepreneurial cultural and industrial growth so as to prepare them to set up and manage their own small units.

Learning Outcomes:

This course will be able to create value. Students are able to create presentations and business plans that articulate and apply financial, operational, organizational, market, and sales knowledge to identify paths to value creation through 1) company formation (for-profit); 2) social innovation (nonprofit).

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
45	N/A	3	N/A	3	N/A	10+5=15	N/A	60	N/A	75	N/A	30	N/A	75+0=75

Unit	Topic	Minimum Class Hours
I	Introduction to the entrepreneur	08
II	Promotion of a Venture	08
III	Entrepreneurial Behaviour	10
IV	Entrepreneurship & Innovation	12
V	Legal and Ethical Considerations	07
Total		45

Detailed Syllabus**Unit I: Introduction to the Entrepreneur****(8 Hours)**

Definition, emergence of entrepreneurial class; Definition and concern of Entrepreneurship, role of social economic environment; classification, Characteristics and importance of entrepreneur; leadership; risk taking; decision making and business planning, Role of entrepreneur

Unit II: Promotion of a Venture**(8 Hours)**

Opportunities analysis; external environmental analysis (economic, social and technological, competitive factors), legal requirements of establishment of a new unit and raising of funds; Venture capital sources and documentation required.

Unit III: Entrepreneurial Behaviour

(10 Hours)

Innovation and entrepreneur (Concept, Creativity, Invention & Innovation, Strategy for Innovation, Effective Commercialization, Innovation and Intellectual Property Rights), entrepreneurial behavior and Psycho- theories.

Entrepreneurial Development Programmes (EDP): EDP, their role, relevance and achievements; role of government in organizing EDP's critical evaluation.

Unit IV: Entrepreneurship & Innovation

(12 Hours)

Overview of project identification, search of a business idea, Identification of project, Business Opportunities, Understanding Design Thinking {concept and scope, key factors of design thinking, benefits, phases (Empathize, Define, Ideate, Prototype, Test)}

Creativity: Creativity, identification creative tools (S-C-A-M-P-E-R), Vertical thinking, lateral thinking, Critical Thinking, Phases of decision making, Critical thinking and objectivity, Applying structured knowledge to unstructured problems, Domain criteria, traditional and out-of-the-box thinking.

Unit V: Legal and Ethical Considerations

(7 Hours)

Legal forms of business organization, ethical Issues and social responsibilities of an entrepreneur.

Recommended Books:

- Vasant, DCSAI; Entrepreneurship, Himalaya Publishing House, 2003.
- Taneja & S.L. Gupta.; Entrepreneurship Development, 2003.
- Pandey, I.M.; venture capital- The Indian Experience, Prentice Hall of India, 2003.

Further readings:

- Tandon B.C, "Environment and Entrepreneur ", Chug publication, Allahabad.
- Srivastava S.B.; A practical guide to industrial entrepreneurs, Sultan Chand & Sons, New Delhi.
- Chandra, Prasana; Project Preparation, Appraisal, Implementation, TMH, New Delhi.



MN-2A (Practical): Entrepreneurship Development Lab

1 Credit | 30 Minimum Class Hours | Semester II

Objectives:

It provides exposure to the students to the entrepreneurial cultural and industrial growth so as to prepare them to set up and manage their own small units.

Learning Outcomes:

This course will be able to create value. Students are able to create presentations and business plans that articulate and apply financial, operational, organizational, market, and sales knowledge to identify paths to value creation through 1) company formation (for-profit); 2) social innovation (nonprofit).

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
N/A	30	N/A	3	N/A	1	N/A	N/A	N/A	25	N/A	25	N/A	10	0+25=25
<u>Marks Distribution of End Semester Practical Examination</u> • Written – 15 Marks • Viva-Voce – 05 Marks • Report File – 05 Marks														

Experiment List

1	Case studies of successful entrepreneurs.
2	Conducting mock interviews: testing initiatives, team spirit and leadership.
3	Conducting meeting: purpose, procedure, participation, physical arrangements, recording and writing of minutes.
4	Share Your Story: Identify area of innovation and prepare a project of design thinking in the area of Your choice and present it through Sketch modeling (Preparing project proposal).
5	Conduct Market survey to know the demand for different products.
6	Presentations by the students (Individuals, Group).



MJ-2 (Theory): Object Oriented Programming with C++

3 Credits | 45 Minimum Class Hours | Semester II

Objectives:

The main objectives of the course are as follows–

- To get a clear understanding of object-oriented concepts.
- To understand the difference between object oriented and procedure oriented programming.
- To understand object-oriented programming through C++.
- To develop real life applications using Object Oriented Programming (OOP) concepts.

Learning Outcomes:

After completion of this course, a student will be able to–

- An understanding of the principles behind the object-oriented development process.
- Competence in the use of object-oriented programming language in the development of small to medium sized application programs.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
45	N/A	3	N/A	3	N/A	10+5=15	N/A	60	N/A	75	N/A	30	N/A	75+0=75

Unit	Topic	Minimum Class Hours
I	Introduction to Object Oriented Programming (OOP) and C++	10
II	Arrays, Strings and Functions	08
III	Classes and Objects, Constructors and Destructors, Operator overloading	12
IV	Inheritance, Polymorphism, and File Handling	15
Total		45

Detailed Syllabus**Unit I: Introduction to Object Oriented Programming (OOP) and C++****(10 Hours)**

Object Oriented Programming: Introduction, OOP vs Procedure Oriented Programming, Basic concepts of OOP - Objects, Classes, Data Abstraction, Data Encapsulation, Data Hiding, Inheritance, Polymorphism, Dynamic binding and Message passing, Benefits of OOP, Applications of OOP.

Introduction to C++: Origins of C++, Structure of C++ program, Tokens, Keywords, Identifiers, Literals/ Constants, Variables, Data Types, Input/ Output in C++, iostream header file, Namespace, main() function, Operators and Expressions in C++, Implicit Conversions, Type casting, Control Structures.

Unit II: Arrays, Strings and Functions

(8 Hours)

Arrays: Declaration, Initialization, One and two-dimensional arrays.

Strings: Declaration, Initialization, Input/Output of strings, String manipulation functions, std::string class in C++.

Functions: Introduction, User defined and Library functions, Function prototypes, Calling functions, Function returning values, Passing arguments, Call by reference, Call by value, Inline functions, Default arguments, Function overloading.

Unit III: Classes and Objects, Constructors and Destructors, Operator overloading

(12 Hours)

Classes and Objects: Defining a class, Members of a class - Data members and Member functions, Private, Public, and Protected visibility modes, Member function definition – inside class and outside class, Declaration of objects, Accessing class members, Arrays of Objects, Objects as Function Arguments, Difference between Structures and Classes.

Constructors and Destructors: Constructors - Characteristics, Declaration and definition, Types of constructors - Default, Parameterized, and Copy constructor, Constructor overloading, Destructors - Characteristics, Declaration and definition.

Operator Overloading: Defining Operator Overloading, Overloading of Unary and Binary Operators, Manipulation of strings using operators, Type Conversions.

Unit IV: Inheritance, Polymorphism, and File Handling

(15 Hours)

Inheritance: Introduction, Base and derived classes, Types of inheritance, Virtual base classes, Abstract classes, Constructors in inheritance, Overriding base class functions.

Polymorphism: Introduction, Compile time and Runtime polymorphism, Pointers to objects, this pointer, Pointers to derived classes, Virtual functions.

Files Handling: Introduction, Stream classes for files, Opening files, File modes, Error handling in file, Detecting end of file-cof(), Sequential input and output-put() and get(), Reading and writing objects-read() and write(), Random Access files.

Recommended Books:

- Balaguruswami, E., **Object Oriented Programming in C++**, Tata McGraw Hill Pub
- Lafore R., **Object Oriented Programming using C++**, Galgotia

Further readings:

- Herbert Schildt, **C++: The Complete Reference**, McGraw Hill
- K R Venugopal, Rajkumar Buyya, T Ravishankar, **Mastering C++**, Tata McGraw Hill



MJ-3 (Theory): Data Structure using C++

3 Credits | 45 Minimum Class Hours | Semester II

Objectives:

Data Structure is considered as one of the fundamental papers towards a more comprehensive understanding of programming and application development. Students are expected to work towards a sound theoretical understanding of Data Structures and also compliment the same with hands on experience with implementation.

The main objectives of the course are as follows–

- To understand the basic concepts of data structures and algorithms.
- To be able to practically implement the data structures like stack, queue, array, lists, tree etc.
- To understand and implement different searching and sorting techniques.

Learning Outcomes:

At the end of the course, students will be able to–

- Understand the need for Data Structures when building application.
- Able to walk through insert and delete for different data structures.
- Ability to calculate and measure efficiency of code.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
45	N/A	3	N/A	3	N/A	10+5=15	N/A	60	N/A	75	N/A	30	N/A	75+0=75

Unit	Topic	Minimum Class Hours
I	Introduction to Data Structure	05
II	Arrays, Searching and Sorting	08
III	Linked list	10
IV	Stacks & Queues	12
V	Trees	10
Total		45

Detailed Syllabus**Unit I: Introduction to Data Structure****(05 Hours)**

Basic concepts and terminology, Importance and applications of data structures, Types of data structures, Overview of operations on data structures: insertion, deletion, traversal, searching, and sorting, Performance Analysis, Performance Measurement.

Unit II: Arrays, Searching and Sorting

(08 Hours)

One-dimensional arrays, Multi-dimensional arrays, Array operations and algorithms: Sorting - Selection sort, Bubble sort, Quick sort, Insertion sort, Merge Sort; Searching- Linear Search and Binary Search.

Unit III: Linked list

(10 Hours)

Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Circular linked list; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion;

Unit IV: Stacks & Queues

(12 Hours)

Stacks: Basic Concepts – Definition and Representation of stacks; Array Implementation of Stack, Operations on stacks; Linked list implementation of Stacks. Applications of stacks – Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack,

Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues (Deque), Priority queues; Operations on Simple queues; Linked list implementation of Queue; Applications of queue.

Unit V: Trees

(10 Hours)

Introduction, Terminology, Representation of Trees, Binary Trees, Properties of Binary Trees, Binary Tree Representations, Binary Tree Traversals, Binary Search Trees: Introduction, Searching a Binary Search Tree, Inserting an Element, Deleting an Element.

Recommended Books:

- Ellis Horowitz and Sartaj Sahni; **Fundamentals of Data Structures in C**; Universities Press
- Reema Thareja; **Data Structures Using C**; Oxford University Press, India

Further Readings:

- Kamathane; **Introduction to Data structures**; Pearson Education
- Y. Kanitkar; **Data Structures Using C**; BPB



MJ (Practical-2): Data Structure and C++ Lab

2 Credits | 60 Minimum Class Hours | Semester II

Objectives:

The main objectives of the course are as follows–

- To get a clear understanding of object-oriented concepts.
- To understand the difference between object oriented and procedure-oriented programming.
- To understand object-oriented programming through C++.
- To develop real life applications using Object Oriented Programming (OOP) concepts.
- To be able to practically implement the data structures like stack, queue, linked-list etc.
- To implement linear and non-linear data structures.
- To understand the different operations of binary search trees.
- To get familiarized to sorting and searching algorithms.

Learning Outcomes:

After completion of this course, a student will be able to–

- Write basic C++ program using control structures, arrays strings etc.
- Demonstrate the concepts of reusability through the use of functions.
- Write object-oriented application using OOP concepts such as encapsulation, abstraction, polymorphism, inheritance etc.
- Use stream classes to manipulate data files.
- Understand the need for Data Structures when building application.
- Write functions to implement linear and non-linear data structure operations.
- Suggest appropriate linear and non-linear data structure operations for solving a given problem.
- Analyze various sorting methods.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
N/A	60	N/A	3	N/A	2	N/A	N/A	N/A	50	N/A	50	N/A	20	0+50=50
<u>Marks Distribution of End Semester Practical Examination</u>														
• Experiments – 30 Marks					• Viva-Voce – 10 Marks					• Practical File – 10 Marks				

Experiment List

Group – ‘A’: C++

Unit I: Introduction to C++ & Control Structures	
1.	Write a program to find roots of a quadratic equation, $ax^2 + bx + c = 0$. Implement following conditions – <ul style="list-style-type: none"> • If the discriminant is positive, then display two distinct real roots. • If the discriminant is zero, then display two equal roots. If the discriminant is negative, then display two distinct complex roots.
2.	Write a program to display Fibonacci series – 0,1,1,2,3,5,8,13,N
3.	Write a menu-based program to check - <ol style="list-style-type: none"> 1. Perfect number 2. Palindrome number 3. Prime number
Unit II: Arrays, Strings and Functions	
4.	Write a program to input N integer elements into an array and swap the largest and lowest element.
5.	Write a program to input order of two matrices and check if it satisfies the condition for addition of the matrices or not. If it satisfies the condition then find the sum of the matrices.
6.	Write a program to input a string, find it's reverse and check whether it is palindrome or not without using any string library functions.
7.	Write a function <i>calculate()</i> which can take maximum three integer arguments x, y, and z, and returns – <ul style="list-style-type: none"> • x^2 if only x is passed. • x^y, if both x and y are passed. • $x^y + x^z$, if all three arguments are passed.
8.	Write a program to show difference between call by value and call by reference.
Unit III: Classes and Objects, Constructors and Destructors, Operator overloading	
9.	Write an object-oriented program to create simple calculator to perform addition, subtraction, multiplication, and division of two numbers. Define appropriate data members and member functions. Your program must use constructor and destructor.
10.	Define a class <i>Number</i> having a private integer data member <i>num</i> and member functions <i>input()</i> and <i>display()</i> to take the input and display the data respectively. Write a program using a friend function to use the class <i>Number</i> to find the sum of two numbers.
11.	Write an object-oriented program to calculate area of square, rectangle and triangle using constructor overloading.

12.	<p>Define a class Tour with following description:</p> <p>Private Members:</p> <table><tr><td>travel_id (integer)</td><td>no_of_adults (integer)</td><td>no_of_kids (integer)</td></tr><tr><td>source (string)</td><td>destination (string)</td><td>distance (float)</td></tr><tr><td>total_fare (float)</td><td></td><td></td></tr></table> <p>Public Members:</p> <ul style="list-style-type: none">A constructor to assign initial value as 0 to all integer and float data members and text “NULL” to all the string data members.calc_fare(): To calculate total_fare as follows – <p>For each adult:</p> <table><tr><td>Fare (Rs.)</td><td>For distance (in K.M)</td></tr><tr><td>-----</td><td></td></tr><tr><td>500</td><td>>=500</td></tr><tr><td>250</td><td><500 and >=300</td></tr><tr><td>150</td><td><300</td></tr></table> <p><i>For each kid the above fare will be 50% of the fare mentioned above.</i></p> <ul style="list-style-type: none">read_data(): To input the values of the data members except total_fare and call calc_fare() function.show_data(): displays the content of all the data members for a Tour.	travel_id (integer)	no_of_adults (integer)	no_of_kids (integer)	source (string)	destination (string)	distance (float)	total_fare (float)			Fare (Rs.)	For distance (in K.M)	-----		500	>=500	250	<500 and >=300	150	<300
travel_id (integer)	no_of_adults (integer)	no_of_kids (integer)																		
source (string)	destination (string)	distance (float)																		
total_fare (float)																				
Fare (Rs.)	For distance (in K.M)																			

500	>=500																			
250	<500 and >=300																			
150	<300																			
13.	Write a program to overload binary Addition (+) operator to find the sum of two complex numbers.																			
14.	Write a program to overload relational operator == to compare two strings.																			
15.	Write a program to demonstrate conversion from Basic to Class type.																			
Unit IV: Inheritance and Polymorphism																				
16.	<p>Create a base class called <i>shape</i>. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called <i>triangle</i> and <i>rectangle</i> from the base <i>shape</i>. Add to the base class, a member function <i>get_data()</i> to initialize base class data members and another member function <i>display_area()</i> to compute and display the area of figures. Make <i>display_area()</i> as a virtual function and redefine this function in the derived classes to suit their requirements.</p> <p>Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles.</p>																			
Unit V: Files and Streams																				
17.	Write a complete employee management file handling program which will perform following operations on binary files - insert records, display record, search records and delete records.																			

Group – ‘B’: Data Structure**Unit I: Searching and Sorting**

- | | |
|----|-----------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Write a function to search an element in the array using binary search. |
| 2. | Write a program to define a function to sort an array using bubble sort or selection sort or quick sort in ascending/ descending order. |

Unit II: Stack

- | | |
|----|-------------------------------------------------------------------------------------------------------------------|
| 3. | Write a menu-based program to implement push, pop and traversal operations on a stack using array or linked list. |
|----|-------------------------------------------------------------------------------------------------------------------|

Unit III: Queue

- | | |
|----|------------------------------------------------------------------------------------------------------------------------|
| 4. | Write a menu-based program to implement insert, delete and traversal operations on a queue using array or linked list. |
|----|------------------------------------------------------------------------------------------------------------------------|

Unit IV: Linked List

- | | |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5. | Write a menu driven program that uses functions to perform the following operations on singly linked list (i) Creation (ii) Insertion (iii) Deletion (iv) Traversal. |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Unit V: Tree

- | | |
|----|---------------------------------------------------------------------------------------------------------|
| 6. | Write a program to create a binary search tree and perform Insertion and different types of traversals. |
|----|---------------------------------------------------------------------------------------------------------|

Note: Additional lab assignments may be included based on topics covered in the theory paper.

