KOLHAN UNIVERSITY

Chaibasa, Jharkhand, India

Syllabus For
Four Year Undergraduate Programme (FYUGP)
Of

Bachelor of Computer Applications (BCA)

With Effect From
Session 2022 - 2026

Table of Contents

Course Structure (Sem- I and II) for
Four Year Undergraduate Programme (FYUGP)
of
Bachelor of Computer Applications (BCA)

Sem.	Paper Code	Paper Title	L–T–P	Credits	Contact Hours
I	CC–1	Language and Communication Skills		6	
	CC–2	Understanding India		2	
	CC–3	Health & Wellness, Yoga Education, Sports & Fitness		2	
	IRC–1	Introductory Regular Courses–1		3	
	IVS–1A	Introductory Vocational Studies–1		3	
	MJ–1	Th: Computer Fundamentals and Introduction to Programming using C	4–0–2	6	60
		Pr: Programming using C Lab			60
II	CC–4	Language and Communication Skills (English)		6	
	CC–5	Mathematical & Computation Thinking Analysis		2	
	CC–6	Global Citizenship Education & Education for Sustainable Development		2	
	IRC–2	Introductory Regular Courses–2		3	
	IVS–1B	Introductory Vocational Studies–2		3	
	MJ–2	Th: Object Oriented Programming using Java	4–0–2	6	60
		Pr: Object Oriented Programming using Java Lab			60
Total Credits				**44**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **CC** (Common Courses), **IRC** (Introductory Regular Courses), **IVS** (Introductory Vocational Studies), **MJ** (Major Disciplinary/Interdisciplinary Courses).

Prepared by,

Mr. Riki Sutradhar
Assistant Professor,
Department of Vocational Courses (B.C.A.),
Jamshedpur Co-operative College, Jamshedpur

Mr. Subodh Kumar
Assistant Professor,
Department of Vocational Courses (B.C.A.),
Jamshedpur Co-operative College, Jamshedpur

Programme Outcomes and Programme Specific Outcomes
of
Bachelor of Computer Applications (BCA)

Programme Outcomes:

1. Acquire knowledge of Computer application theory and algorithm principles in the design and modeling of computer based system.

2. Understand the computing concepts and their applications using the acquired board based knowledge.

3. To provide thorough understanding of nature, scope and application of computer and computer languages.

4. Identify and analyze software application problems in multiple aspect including coding, testing and implementation in industrial applications.

5. The program prepares the young professional for a range of computer applications, computer organization, and techniques of Computer Networking, Software Engineering, Web development, Database management and Advance Java.

Programme Specific Outcomes:

1. To pursue further studies to get specialization in Computer Science and Applications, Economics, Mathematics, Business Administration.

2. To pursue the career in corporate sector can opt for MBA, MCA, etc.

3. To Work in the IT sector as programmer, system engineer, software tester, junior programmer, web developer, system administrator, software developer, etc.

4. To work in public sector undertakings and Government organizations.

5. Ability to understand the changes or future trends in the field of computer application.

6. Encouraging students to convert their start-up idea to reality by implementing.

7. Students will able to understand, analyze and develop computer programs in the areas related to algorithm, system software, web design and networking for efficient design of computer–based system.

Semester – I

Sem.	Paper Code	Paper Title	L–T–P	Credits	Contact Hours
I	CC–1	Language and Communication Skills		6	
	CC–2	Understanding India		2	
	CC–3	Health & Wellness, Yoga Education, Sports & Fitness		2	
	IRC–1	Introductory Regular Courses–1		3	
	IVS–1A	Introductory Vocational Studies–1		3	
	MJ–1	Th: Computer Fundamentals and Introduction to Programming using C	4–0–2	6	60
		Pr: Programming using C Lab			60
Total Credits				**22**	

MJ–1 (Th): Computer Fundamentals and Introduction to Programming using C

4 Credits | 60 Minimum Class Hours | Semester I

Objectives:

The objective of the course is to introduce the fundamentals computer system and C programming language.

Learning Outcomes:

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

- Understand and use the process of abstraction using a programming language such as 'C'.
- Analyze step by step and develop a program to solve real world problems.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Fundamentals of Computer	03
II	Data Representation	03
III	C Fundamentals	10
IV	Control Structures and C Preprocessor	12
V	Arrays, Strings, Pointers, and Functions	20
VI	Structures, Unions and File Handling	12
	Total	**60**

Detailed Syllabus

Unit I: Fundamentals of Computer

***Structure and Working of Computer*:** Functional Block Diagram of Computer {Central Processing Unit (Control Unit and Arithmetic Logic Unit), Input Unit, Output Unit, Memory Unit (Primary Storage Unit and Secondary Storage Unit), Bus Structure}.

***Input/Output Devices*:** Input Devices (Keyboard, Mouse, Scanner, MICR, OMR), Output Devices (VDU, Printers, Plotter, Projector, Speakers).

***Computer Memory*:** Primary Memory (RAM, ROM, Register, Cache Memory, Virtual Memory, etc.), Secondary Memory (Magnetic Tape, Magnetic Disk, Optical Disk, Floppy Disk, External Hard Drive, Solid State Drive, USB Flash Drive, etc.).

***Programming Language*:** Low Level Language (Machine Language, Assembly Language), High Level Language (Procedural–Oriented Language, Problem–Oriented Language, Natural Language), Pseudo Code, Flowchart.

***Computer Software*:** Introduction, Categories of Computer Software {System Software (Operating System, Basic Input/Output System, System Utility, Device Driver, Programming Software, Firmware, etc.) and Application Software (Word Processor, Spreadsheet Software, Database Software, Multimedia Software, Application Suit, Internet Browser, Email Program, etc.)}.

Unit II: Data Representation

***Number System*:** Binary Number System, Octal Number System, Decimal Number System, Hexadecimal Number System, Conversion from One Number System to Another, Arithmetic Operations (Addition, Subtraction, Multiplication, and Division) on Binary Number System.

***Fixed–Point Number (i.e., Integer) Representation*:** Unsigned Integers, Signed Integers (Sign–Magnitude, 1's Complement, and 2's Complement Representation).

***Floating–Point Number Representation*:** 32–Bit Single–Precision Floating–Point Numbers, 64–Bit Double–Precision Floating–Point Numbers.

***Character Encoding*:** Bit, Byte, Word, BCD, EBCDIC, ASCII, ANSI, Unicode, UTF, ISCII.

Unit III: C Fundamentals

History, Structures of 'C' Programming, Function as Building Blocks, Character Set, Tokens, Keywords, Identifiers, Variables, Constant, Data Types, Comments.

***Operators*:** Types of Operators, Precedence and Associativity, Expression, Statement and Types of Statements.

Built-in Functions: Console I/O Functions {scanf( ), printf( ), getch( ), getche( ), getchar( ), gets( ), putch( ), putchar( ), puts( )}, Character functions {isalpha( ), isdigit( ), isalnum( ), isspace( ), islower( ), isupper( ), isxdigit( ), iscntrl( ), isprint( ), ispunct( ), isgraph( ), isblank( ), tolower( ), toupper( )}.

Unit IV: Control Structures and C Preprocessor

***Control Structures*:** Sequence Structure, Selection Structure (if Statement, if–else Statement, if-else if-else Statement, Nested if–else Statement, switch–case Statement), Loop Structure (while, do–while, for Loop, Nested Loop), Other Statements (break, continue, goto).

***C Preprocessor*:** Types of C Preprocessor Directives, Comparison of Macros with Functions, File Inclusion.

Unit V: Arrays, Strings, Pointers, and Functions

***Arrays*:** One–Dimensional Arrays (Definition, Declaration, Initialization, Accessing and Displaying Array Elements, Passing Array to a Function), Two–Dimensional Arrays (Definition, Declaration, Initialization, Accessing and Displaying Array Elements).

***Strings*:** Definition, Declaration, Initialization, Standard Library Functions {strlen( ), strlwr( ), strupr( ), strcat( ), strncat( ), strcpy( ), strncpy( ), strcmp( ), strncmp( ), strcmpi( ), stricmp( ), strnicmp( ), strdup( ), strchr( ), strrchr( ), strstr( ), strset( ), strnset( ), strrev( )}.

***Pointers*:** Definition, Declaration, Initialization, Indirection Operator, Address of Operator, Operations on Pointers, Array of Pointers, Dynamic Memory Allocation.

***Functions*:** Declaration and Definition, Function Call, Types of Function, Parameter Passing (Call by Value, Call by Reference), Scope of Variables, Storage Classes (Automatic, Register, Extern, Static Variables), Nested Function, Recursive Function.

Unit VI: Structures, Unions and File Handling

***Structures*:** Definition, Declaration, Initialization, Accessing Structure Elements, Array of Structures, Pointers and Structures, Passing Structures to a Function.

***Union*:** Definition, Declaration, Initialization, Accessing Structure Elements, Differentiate between Structure and Union, Enumerated Data Type.

***File Handling*:** Introduction, Defining and Opening a File, Closing a File, Input/Output Operations on Text and Binary Files, Error Handling During I/O Operation, Random Access to Files, Standard Function {fopen( ), fclose( ), feof( ), fseek( ), rewind( )}, Using Text Files {fgetc( ), fputc( ), fprintf( ), fscanf( ), etc.}.

Recommended Books:

- Floyd, T. L., **Digital Fundamentals** (Fifth Edition), New Delhi: Pearson Education, 2002
- Hamacher, V. C.; Z. G. Vranesic; S. G. Zaky, **Computer Organization** (Fourth Edition), New Delhi: Tata McGraw-Hill, 1996
- Rajaraman V., **Computer Programming in C** (Second Edition), New Delhi: Tata McGraw-Hill Publication, 1992
- Kanetkar Y., **Let Us C** (Third Edition) ,New Delhi: BPB Publications, 1999
- Gottfried, B. S., **Theory and Problems of Programming with C**, New Delhi: Tata McGraw-Hill Publication, 1997
- Balaguruswamy E. **Programming in ANSI C** (Second Edition), New Delhi: Tata McGraw-Hill Publication, 1992

Further readings:

- Dennis Ritchie, **The C Programming Language**, New Delhi: Pearson Education
- Forouzah, Ceilberg Thomson, **Structured Programming Approach Using C**, Learning Publication
- Deitel & Deitel, **C How To Program,** New Delhi: Prentice Hall India, 1996
- R. B. Patel, **Fundamental of Computers and Programming in C**, Khanna Book Publishing Compa ny PVT. LTD. Delhi, India, 1st edition, 2008

MJ–1 (Pr): Programming using C Lab

2 Credits | 60 Minimum Class Hours | Semester I

Objectives:

This course helps the students in understanding a powerful, portable and flexible structured programming language which is suitable for both systems and applications programming. It is a robust language which contains a rich set of built–in functions and operators to write any complex program.

Learning Outcomes:

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

- Develop modular, efficient and readable C programs by hands–on experience.
- Interpret good profound knowledge in C programming language and enable them to build programs using Control Structures, Arrays, Strings, Pointers, Functions, Structures, Unions and File Handling to solve the real world problems.
- Illustrate memory allocation to variables dynamically and perform operations on text and binary files.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Experiment List

Unit I: C Fundamentals

1. Write a program to evaluate the arithmetic expression ((A + B / B * D - E) * (F – G)). Read the values of A, B, C, D, E, F, and G from the standard input device.
2. Write a program to check whether a number is even or odd using ternary (or, conditional) operator.
3. Write a program to perform addition of two numbers without using '+' operator.
4. Write a program to find ASCII value of a character entered by user.
5. Write a program to find quotient and remainder by a division process.
6. Write a program to find the size of int, float, double and char data type.
7. Write a program to swap two numbers without using temporary variable.

Unit II: Control Structures and C Preprocessor

1. Write a program to find the largest number among three numbers.
2. Write a program to find all roots of a quadratic equation.

3. Write a program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, *, /, % and use "switch–case" statement).

4. Write a program to find factorial of a given number.

5. Write a program to generate Fibonacci series up to N^{th} term.

6. Write a program to check whether a number is palindrome or not.

7. Write a program to check whether a number is prime or not.

8. Write a program to check whether a N digits number is Armstrong number or not.

9. Define a macro with one parameter to compute the volume of a sphere. Write a program using this macro to compute the volume for spheres of radius 5, 10 and 15 meters.

10. Define a macro that receives an array and the number of elements in the array as arguments. Write a program for using this macro to print the elements of the array.

11. Write symbolic constants for the binary arithmetic operators +, -, *, and /. Write a program to illustrate the use of these symbolic constants.

Unit III: Arrays, Strings Pointers, and Functions

1. Write a program to find the sum of all elements, average of all elements, and the second largest integer in a "One–Dimensional" integer array.

2. Write a program to swap first and last element of a "One–Dimensional" integer array.

3. Write a program that lets the user perform arithmetic operations on two "Two–Dimensional" integer arrays. Your program must be menu driven, allowing the user to select the operations (e.g., Press 1 for Addition and Press 2 for Multiplication).

4. Write a program to reverse a string.

5. Write a program to count number of vowels, consonants and spaces in a given string.

6. Write a program that lets the user perform string operations on standard library functions. Your program must be menu driven, allowing the user to select the operations (e.g., Press 1 to demonstrate the usage of function "strlen( )", Press 2 to demonstrate the usage of function "strlwr( )", Press 3 to demonstrate the usage of function "strupr( )", Press 4 to demonstrate the usage of function "strcat( )", and so on).

7. Write a program to concatenate two strings using pointer.

8. Write a program to find the length of a string using pointer.

9. Write a program to read and print an integer array. The program should input total number of elements (limit) and elements in array from user. Use dynamic memory allocation to allocate (i.e., "malloc()" function) and deallocate (i.e., "free()" function) array memory.

10. Write a program to read an integer array and find maximum and minimum number in the array. The program should input total number of elements (limit) and elements in array from user. Use

dynamic memory allocation to allocate (i.e., "calloc()" function) and deallocate (i.e., "free()" function) array memory.

11. Write a program to read and print an integer array. The program should input total number of elements (limit) and elements in array from user. Use dynamic memory allocation to allocate (i.e., "calloc()" function) and deallocate (i.e., "free()" function) array memory. After that use "realloc()" function to alter the size of existing allocated memory blocks for the integer array and print the array.

12. Write program that use function to return the greatest common divisor of two given integers.

13. Write a program to print the transpose of a given matrix using function.

14. Write a program to generate Fibonacci series up to N^{th} term using recursive function.

Unit IV: Structures, Unions and File Handling

1. Write a program that lets the user perform arithmetic operations on two complex numbers. Define a structure that will hold the data for a complex number. Your program must be menu driven, allowing the user to select the operations (+, -, and *) and input the complex numbers. Furthermore, your program must consist of following functions:

(i) Function "**showChoice()**": This function shows the options to the user and explains how to enter data.
(ii) Function "**add()**": This function accepts two complex number structures as arguments and returns a complex number structure with the sum of the two complex numbers.
(iii) Function "**subtract()**": This function accepts two complex number structures as arguments and returns a complex number structure with the difference of the two complex numbers.
(iv) Function "**multiply()**":This function accepts two complex number structures as arguments and returns a complex number structure with the product of the two complex numbers.

2. Create a union named '**Book**' containing '**book_id**', '**title**', '**author_name**' and '**price**'. Write a program to pass the union as a function argument and print the book details.

3. Write a program to display the contents of a text file.

4. Write a program to copy the contents of one file to another file.

5. Write a program to create a text file named "**MyInfo.txt**", open it, type-in some information about yourself. Read and count the number of characters in the file.

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