



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

Serial Number	Topics	Page Number
1	Course Structure (Sem- I and II) for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)	03 – 03
2	Programme Outcomes and Programme Specific Outcomes of Bachelor of Computer Applications (BCA)	04 – 04
3	Semester – I	05 – 11
4	Semester – II	12 – 19

**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
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
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 Pr: Programming using C Lab	4-0-2	6	60 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(), putchar(), putchar(), puts()}, Character functions {isalpha(), isdigit(), isalnum(), isspace(), islower(), isupper(), isxdigit(), isctrl(), 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(), strlen(), strcpy(), strcat(), strncat(), strncpy(), strncmp(), strncmp(), strcmpi(), stricmp(), strnicmp(), strdup(), strchr(), strchr(), strstr(), 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 Company 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 Nth 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.



Semester – II

Sem.	Paper Code	Paper Title	L–T–P	Credits	Contact Hours
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
Pr: Object Oriented Programming using Java Lab		60			
Total Credits				22	

MJ-2 (Th): Object Oriented Programming using Java

4 Credits | 60 Minimum Class Hours | Semester II

Objectives:

The course will introduce students to object oriented programming using Java. It assumes that students know the basics of scalar types (integers, strings and booleans) and fundamental control structures in procedural programming (loops, assignment statements, conditional expressions). It will focus on more sophisticated features such as design of classes, inheritance, interfaces, packages and APIs. It will also cover the basic principles of event handling, multithreading, exception handling, swing programming and JDBC.

The main objectives of the course are as follows–

- Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc.
- Understand the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc. and exception handling mechanisms.
- Understand the principles of inheritance, packages and interfaces.
- Gain knowledge in the concepts of exception handling, swing and JDBC.

Learning Outcomes:

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

- Analyze the logic of a given problem.
- Use branching control statements and iterative control statements.
- Achieve Multiple Inheritance using Interface.
- Understand Java Swing and JDBC to design application.

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	Introduction and Fundamentals of Java	12
II	Classes, Objects and Methods	12
III	Arrays, Strings, and Wrapper classes	08
IV	Packages, Multithreading and Exception Handling	08
V	Managing Input/ Output Files in Java, Java Collection Framework	08
VI	Event and GUI programming, JDBC	12
Total		60

Detailed Syllabus

Unit I: Introduction and Fundamentals of Java

Concepts of Object Oriented Programming, Benefits of OOP, History and Features of Java, C/C++ vs Java, Java Runtime Environment, Java Virtual Machine, Java Development Kit, Java Compiler And Interpreter, Java Program Structure ,Implementing a Java Program, Tokens- Keywords, Identifiers, Constants, Data Types- Primitive & Non- Primitive Data Types, Object Reference

Types, Strings, Variable declaration & initialization, Types of variables such as local, instance, and static variables, Java Operators and Expressions, Type Conversion in Expressions, Operator Precedence and Associativity, Mathematical Functions, Input/ Output in Java, Command Line Arguments.

Control Structures: Decision Making and Branching Statements - if Statement, The if... else Statement, Nesting of if ... else Statements, The else if Ladder, The switch Statement, Iteration or Loop Statements - while Loop, do ... while Loop, for Loop, for each Loop, Labeled Loop, Nested Loops, Jump Statements - break and continue Statements, return Statement.

Unit II: Classes, Objects and Methods

Defining a Class, Fields declaration, Method declaration, Creating object, Anonymous object in Java, Accessing class members, Access or Visibility Modifier, this keyword, Method overloading, Constructors, Constructor overloading, static members.

Inheritance and Polymorphism: Inheritance Basics, Super and Sub class, Types of Inheritance, Overriding methods, super keyword, final (variables, methods and classes), Static and Dynamic Binding, Abstract methods and classes.

Interfaces: Defining Interfaces, Implementing Interfaces, Extending Interfaces, Accessing Interface variable, Multiple Inheritance through Interfaces, Classes V/s Interfaces.

Unit III: Arrays, Strings and Wrapper classes

One Dimensional Arrays, Declaration, Creation, Initialization of Arrays, Two Dimensional Arrays, String class and its methods, Wrapper Classes.

Unit IV: Packages, Multithreading and Exception Handling

Package: Introduction, Java API Packages, Using System Packages, Naming Conventions, Creating Packages, Accessing Packages, Using a Package, Adding a Class to a Package, Hiding Classes.

Multithreading: Introduction, Creating Threads, Extending the Thread Class, Stopping and Blocking a Thread, Life Cycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, and Synchronization, Deadlock.

Exception Handling: Introduction, Types of Errors, Exceptions, Syntax of Exception Handling Code, Multiple Catch Statements, Using finally Statement.

Unit V: Managing Input/ Output Files in Java, Java Collection Framework

Streams in Java, Stream Classes – Character (File Reader & Writer classes) & Byte Streams (File I/O Stream Classes), Using the File class, Creating Files, Reading/Writing Characters and Bytes to File, Random Access Files.

Java Collection Framework: Introduction, Collection Framework Interfaces (Set, List, Queue, Iterator) and Classes (ArrayList, Vector, LinkedList, Stack).

Unit VI: Event and GUI programming, JDBC

Event Handling: Event Classes, Sources of Events, Event Listeners, Key and Mouse Event Handling.

Swings: Architecture, Components of Swing- JLabel, JButton, JCheckBox, JRadioButton, JList, JComboBox, JTextField, JTextArea, JPanel, JFrame etc., Working With Graphics, Working with Colour, Adding And Removing Controls, Responding To Controls, Layout Managers (Flow Layout, Grid Layout, Card Layout, Border Layout), Handling Events.

Database Connectivity using JDBC: JDBC Architecture, JDBC Drivers, Using Connection, Statement & Resultset Interfaces for Manipulating Data with Databases.

Recommended Books:

- E. Balagurusamy; **Programming with JAVA**; McGraw Hill, New Delhi
- Joel Murach, Michael Urban; **Murach's Beginning Java with Net Beans**; SPD

Further readings:

- Herbert Schildt; **Java: The Complete Reference**; McGraw Hill
- Raj Kumar Buyya; **Object Oriented Programming with JAVA**; McGraw Hill
- Ken Arnold, James Gosling; **The Java Programming Language**; Addison Wisely
- Wiley; **Java 6 Programming Black Book**; Kogent Learning Solutions



MJ-2 (Pr): Object Oriented Programming using Java Lab

2 Credits | 60 Minimum Class Hours | Semester II

Objectives:

The main objectives of the course are as follows:

- To teach the students basics of JAVA programs and its execution.
- To make the students learn concepts like packages and interfaces.
- To teach applications using Console I/O and File I/O, GUI and JDBC applications.

Learning Outcomes:

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

- Use Java compiler and other platform to write and execute java program.
- Understand and Apply Object oriented features and Java concepts.
- Apply the concept of multithreading and implement exception handling.
- Access data from a Database with java program.
- Develop applications using Console I/O and File I/O, GUI applications.

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: Data Types, Operators and Expressions, Selection and Loop Statements	
1.	Write a program to input and display different types of data values using Scanner or Stream class.
2.	Write a program to create a simple calculator which can perform basic arithmetic operations like addition, subtraction, multiplication or division, exponent (x^y), log and square root depending upon the user input.
3.	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.
4.	Write a program to input length of three sides of a triangle. Then check if these sides will form a triangle or not. If sides form a triangle then display the type of the triangle with its area and perimeter.
5.	Write a program to find factorial of list of numbers reading input as command line argument.
6.	Write a menu based program to check Prime, Armstrong, Automorphic and Krishnamurty number.

7.	<p>Write a menu based program to convert following –</p> <ul style="list-style-type: none"> • Decimal to Binary Number • Binary to Decimal Number
----	---

Unit II: Arrays, Strings and Wrapper Classes

1.	Write a java program to check whether the elements of an array are sorted or not. If sorted, display the order, otherwise sort elements in ascending order or descending order as per user's choice.
2.	Write a program to input order of two matrices and check if it satisfies the condition for product of the matrices or not. If it satisfies the condition then find the product of the matrices. <i>(Hint: Two matrices can be multiplied if and only if they satisfy the following condition: The number of columns present in the first matrix should be equal to the number of rows present in the second matrix.)</i>
3.	Write a program to input a string and display number of vowels and consonant in each word. For example if the string is "Kolhan University Chaibasa", then the output will be – Kolhan: Vowels = 2, Consonants = 4 University: Vowels = 4, Consonants = 6 Chaibasa: Vowels = 4, Consonants = 4
4.	Write a program to count number the of palindrome words in a sentence and display the longest palindrome word.
5.	Write a program to implement different methods of wrapper class.
6.	Write a program to convert primitive to wrapper class and wrapper to primitive.

Unit III: Classes, Objects and Methods

1.	Write a program in Java with class Rectangle with the data fields width, length, area and color .The length, width and area are of double type and color is of string type .The methods are set_ length () , set_ width () , set_ color(), and find_ area (). Create two object of Rectangle and compare their area and color. If area and color both are same for the objects then display "Matching Rectangles" otherwise display "Non matching Rectangle".
2.	Create a class Account with two overloaded constructors. First constructor is used for initializing, name of account holder, account number and initial amount in account. Second constructor is used for initializing name of account holder, account number, addresses, type of account and current balance. Account class is having methods Deposit (), Withdraw (), and Get_Balance(). Make necessary assumption for data members and return types of the methods. Create objects of Account class and use them.
3.	Write a Java program to create a shape class and derive, square and circle classes from shape class. Define appropriate constructor for all the three classes. Define a method Area() to calculate area of circle and square in respective class. Assume PI = 3.14 and declare it as a final variable in circle class.

4.	Define an Employee class with suitable attributes having getSalary() method, which returns salary withdrawn by a particular employee. Write a class Manager which extends a class Employee, override the getSalary() method, which will return salary of manager by adding traveling _allowance, house rent allowance etc. Use default and parameterized constructors to initialize data.
5.	Write a java program which creates an interface having 2 methods add () and sub(). Create a class which implements the above interface for addition and subtraction of two numbers respectively.
7.	Write a program to demonstrate the multiple inheritance using interfaces.

Unit IV: Packages, Multithreading and Exception Handling

1.	Write a program to make a package Balance in which has Account class with Display_Balance method in it. Import Balance package in another program to access Display_Balance method of Account class.
2.	Write a java program to implement thread life cycle.
3.	Write a program that creates 2 threads - each displaying a message (Pass the message as a parameter to the constructor). The threads should display the messages continuously till the user presses Ctrl+C.
4.	Write a Java program to use the try, catch and finally block to handle the exception.
5.	Write a program to illustrate the throws keyword in Java.
6.	Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.

Unit V: Managing Input/ Output Files in Java, Java Collection Framework

1.	Write a java program to copy the contents from one file to other file.
2.	Write a java program to read the student data from user and store it in the binary file.
3.	Write a program to add, retrieve and remove the element from the ArrayList.
4.	Write a program to implement LinkedList and perform different operations on it.
5.	Write a program to implement push() and pop() operation on Stack in Java using java Collection.

Unit VI: Event and GUI programming, JDBC

1.	Java program to create a simple calculator with basic operations such as +, -, /, * using java swing elements. Use appropriate layout.
----	--

2.	Write a program to create a combo box which includes list of subjects and radio buttons to show different colour options. Display the selected subject in the text field with selected colour.
3.	Develop an application (GUI or Console based) to connect to a database created in MYSQL/ MS–ACCESS/ SQL–SERVER/ ORACLE and Perform basic operations of Selection, Insertion, Modification and Deletion on the database using JDBC.

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

