



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW**

Course Structure and Syllabus

For

**BACHELOR OF COMPUTER APPLICATION
(BCA)**

2nd Year

as per

NEP-2020

(To be effective from the session 2024-2025)

[Handwritten signatures]

BACHELOR OF COMPUTER APPLICATION (BCA)

YEAR: SECOND, SEMESTER-III

(To be effective from the session 2024-2025)

S. No.	Paper Code	Subject	Periods			Evaluation Scheme				Sub Total	Credit
			L	T	P	Sessional Exam			Exam ESE		
						CT	TA	Total			
1.	NBCA-301	Object Oriented Programming Using Java	3	1	0	20	10	30	70	100	4
2.	NBCA-302	Software Engineering	3	1	0	20	10	30	70	100	4
3.	NBCA-303	Computer Architecture	3	1	0	20	10	30	70	100	4
4.	NBCA-304	Python Programming	3	1	0	20	10	30	70	100	4
5.	NBCA-305	Accounting and Financial Management	3	0	0	20	10	30	70	100	3
Practical											
6.	NBCA-306P	Java Lab	0	0	3		20	20	30	50	2
7.	NBCA-307P	Python Programming Lab	0	0	3		20	20	30	50	2
8.	NBCA-308P	Industrial Training viva-voce	0	0	2		20	20	30	50	1
9.	NBCA-GP	General Proficiency	-	-	-		-	--	-	50	
		Total	15	4	8					650	24

Handwritten signatures and initials

BACHELOR OF COMPUTER APPLICATION (BCA)

YEAR: SECOND, SEMESTER-IV

(To be effective from the session 2024-2025)

S. No.	Paper Code	Subject	Periods			Evaluation Scheme				Sub Total	Credit
			L	T	P	Sessional Exam			Exam ESE		
						CT	TA	Total			
1.	NBCA-401	Advance Java Technology	3	1	0	20	10	30	70	100	4
2.	NBCA-402	Design and Analysis of Algorithm	3	1	0	20	10	30	70	100	4
3.	NBCA-403	Web Design Concepts	3	1	0	20	10	30	70	100	4
4.	NBCA-404	Computer Graphics	3	1	0	20	10	30	70	100	4
5.	NBCA-405	Managerial Economics	3	0	0	20	10	30	70	100	3
Practical											
6.	NBCA-406P	Advance Java Technology Lab	0	0	3		20	20	30	50	2
7.	NBCA-407P	Web Design Lab	0	0	3		20	20	30	50	2
8.	NBCA-408P	Computer Graphics Lab	0	0	2		20	20	30	50	1
9.	NBCA-GP	General Proficiency	-	-	-		-	--	-	50	
		Total	15	4	8					650	24

Note:

1. After Examination of Semester-IV, the Industrial Project based Training (04 to 06 weeks) to be conducted during summer break, and will be assessed during V semester (BCA Degree Programme). The training may be carried out at some industrial unit or under the guidance of faculty member of the institution.
2. If the student leaves the programme after completing Semester-IV successfully, student will be awarded a **Diploma in Computer Application**.

OBJECT ORIENTED PROGRAMMING USING JAVA

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understand the fundamentals of object oriented programming.
- Understand java programming basics.
- Use inheritance, polymorphism, arrays, threads, packages, and exception handling.

Unit-I 08
Object-Oriented Analysis: Introduction to Object Oriented Concepts, Object Oriented Analysis Modeling, Data Modeling, Origin of Object-Oriented Design, Object Oriented Design Concepts, Object Oriented Design methods, Class and object definition, Refining operations, Program Components and Interfaces, Annotation for Object-Oriented Design, Implementation of Detail Design.

Unit-II 08
Java Basic : JAVA environment, JAVA program structure, Tokens, Statements, JVM, Constant and Variables, Data Types, Declaration of variables, Scope of variables, Symbolic constants, Type Casting. Operators: Arithmetic, Relational, Logical assignments, Increment and Decrement, Conditional, Bitwise, Special, Expressions and its evaluation. Object and Class Concept: Defining a Class, Adding variables and Methods to classes, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members, and Nesting of Methods.

Unit-III 08
Inheritance: Extending a Class, Overriding Methods, Final Variables and Methods, Final Classes, Finalize Methods, Abstract Methods and Classes, Visibility Control. Arrays: One Dimensional and Two Dimensional, Strings, Vectors, Wrapper Classes. Interface: Defining Interface, Extending Interface, Implementing Interface, Accessing Interface Variable.

Unit-IV 08
Arrays: One Dimensional and Two Dimensional, Strings, Vectors, Wrapper Classes. Exception Handling: Concepts of Exceptions, Types of Exception, Try and Catch keyword, Nested Try and Catch. Package: System Packages, Using System Package, Adding a Class to a Package, Hiding Classes.

Unit-V 08
Exception Handling: Concepts of Exceptions, Types of Exception, Try and Catch keyword, Nested Try and Catch. Threads: Creating Threads, Extending Threads Class, Stopping and Blocking a Thread, Life Cycle of a Thread, Using Thread Methods, Thread Exceptions. Package: System Packages, Using System Package, Adding a Class to a Package, Hiding Classes.

Text Book:

1. E. Balagurusamy, "Programming in Java", TMH Publications.

Reference Books:

1. Peter Norton, "Peter Norton Guide to Java Programming", Techmedia Publications.
2. Naughton, Schildt, "The Complete Reference JAVA 2", TMH.

SOFTWARE ENGINEERING

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understand the basic concepts of software engineering.
- Understand the requirement analysis and importance of SRS documentation.
- Understand the design of software product.
- Understand various testing techniques and maintenance of software product.

Unit-I 08
Introduction: Software components, software characteristics, software crisis, software engineering processes, similarity and differences from conventional engineering processes, and software quality attributes. SDLC Models: Water fall model, prototype model, spiral model, evolutionary development models, iterative enhancement models, and agile software development model.

Unit-II 08
Software Requirement Specifications: Requirement engineering process, feasibility study, information modeling, data flow diagrams, entity relationship diagrams, decision tables, SRS document, and IEEE standards for SRS. SQA: Verification and validation, SQA plans, software quality frameworks, ISO 9000 models, and SEI-CMM model.

Unit-III 08
Software Design: Basic concept, architectural design, low level design: modularization, design structure charts, pseudo codes, flow charts, coupling and cohesion measures, design strategies. Software Measurement and Metrics: Halstead's software science, function point (FP) based measures, and cyclomatic complexity measures: Control flow graphs.

Unit-IV 08
Software Testing: Testing objectives, unit testing, integration testing, acceptance and regression test, testing for functionality and performance. Top-down and bottom-up testing strategies: test drivers and test stubs, structural testing (white box testing), functional testing (black box testing), test data suit preparation, alpha and beta testing of products, Static testing strategies.

Unit-V 08
Software Maintenance and Software Project Management: Software as an evolutionary entity, need for maintenance, categories of maintenance, cost of maintenance, software re-engineering, reverse engineering, software configuration management activities, change control process, software version control, an overview of CASE tools, estimation of cost, efforts, schedule/duration, and constructive cost models (COCOMO).

Text Book:

1. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.

Reference Books:

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
3. Pankaj Jalote, Software Engineering, Wiley
4. Carlo Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication.

COMPUTER ARCHITECTURE

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understand in depth about architecture of computer system.
- Analyse and understand the concepts of parallel processing.
- Understand the concepts of pipeline processing.

Unit-I

08

Introduction: Functional units of digital computer system and their interconnections, introduction to Arithmetic and logic unit, basics of control unit, fundamentals of memory, input/output and peripheral devices introduction. Difference between computer organization and architecture.

Unit-II

08

Architecture: Different components of a computer, Classification of computers (Flynn's and Feng's Classification), Introduction to RISC and CISC architecture, Comparison between Pipelining and Parallelism, Computer instruction types: formats, instruction cycles & sub-cycles, micro operations and execution of complete instruction.

Unit-III

08

Basic Concepts of Parallel Processing: Concept of parallelism and Network Models (Mesh, Pyramid, Butterfly and Hypercube models), Performance metrics & measures and speed up performance laws, Parallel Random-Access Machine Models (PRAM models).

Unit-IV

08

Pipeline Processing: Principle of pipelining, general structure of pipelines, classification of pipeline processors, general pipeline and reservation tables. Principle of Designing pipelined Processor: pipeline instruction execution, pre-fetched buffer, internal forwarding and register tagging, hazard detection & resolution. Pipeline Scheduling Theory.

Unit-V

08

Program Partitioning & Scheduling: Grain size & latency, Grain packing, scheduling and static multiprocessor scheduling. Program Flow Mechanism: control flow vs data flow, demand driven mechanism and comparison of flow mechanism.

Text Book:

1. John P Hayes "Computer Architecture and organization" McGraw Hill
2. Dezso Sima, Terence Fountain and Peter Kacsuk "Advanced Computer Architecture" Pearson Education
3. Kai Hwang "Advanced Computer Architecture" TMH

Reference Books:

1. Linda Null, Julia Lobur- The Essentials of Computer Organization and Architecture, 2014, 4th Edition.
2. Rao, P.V.S. Prospective in Computer Architecture" Prentice Hall of India
3. William Stallings "Computer Organization and Architecture" Pearson
4. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization" McGraw Hill Fifth International Edition

(Handwritten signatures and initials)

PYTHON PROGRAMMING

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understand the basic concept of python.
- Understand the variable, data type, loop, and properties of python.
- Understand the concept of strings and its associated functions.
- Understand the object-oriented concept in python.
- Apply knowledge of python on file using pandas and numpy.

Unit-I **08**

Basics of Python: The programming cycle for python, python IDE, interacting with python programs, elements of python, variables, data types, type conversion. Expressions, assignment statement, arithmetic operators, operator precedence and Boolean expression.

Unit-II **08**

Conditional program execution: Conditional statement, looping, control statements: if, if else, nested if else, for loop, while loop, nested loop. **Lists:** Introduction, properties, accessing list, operations, working with functions and methods. **Tuple:** Introduction, properties, accessing tuple, operations, working with functions and methods. **Dictionaries:** Introduction, properties, accessing values in dictionaries, working with functions and methods.

Unit-III **08**

Strings and Functions: String manipulation: accessing strings, basic operations, string slices. Functions: definition, calling a function, types of functions, function arguments, anonymous functions, global and local variables.

Unit-IV **08**

OOP Concepts: Classes and objects, definition, creating classes, instance methods, new style class, attributes, inheritance, polymorphism, exception classes, custom exception, overloading, overriding and data hiding.

Unit-V **08**

File handling in Python: Opening and closing file, reading and writing files. Searching- Linear and Binary search, sorting-merge sorting, insertion sort, and selection sort.

Text Book:

1. Allen B. Downey, Think Python: How to think like a Computer Scientist“, 2nd edition, Updated for Python 3, Shroff/O’Reilly Publishers, 2016.
2. Guido van Rossum and Fred L. Drake Jr, “An Introduction to Python- Revised and updated for Python 3.2”, Network Theory Ltd, 2011.
3. John V Guttag, “Introduction to computation and programming using Python”, Revised and expanded Edition, MIT Press, 2013.

Reference Books:

1. Robert Sedgewick, Kevin Wayne, Robert Dondero, “Introduction to Programming in Python: An Inter-disciplinary Approach”, Pearson India Education Services Pvt Ltd, 2016.
2. Timothy A. Budd, “Exploring Python”, Mc-Graw Hill Education (India) Private Ltd, 2015
3. Kenneth A. Lambert, “Fundamentals of Python: First Programs”, CENGAGE Learning, 2012.

Course Outcomes (COs):

At the end of this course students will be able to:

- Write programs in java language.
- Implement interface and package.
- Implement Method Overloading and Method Overriding. Handle exceptions in java.

LIST OF PRACTICALS

Note: - At least ten experiments are to be conducted. Perform practical using JAVA language.

1. Write a program in java which prints your name using command line arguments.
2. Write a program in java which enters three number using command line arguments and print sum and average of the number
3. Write a program to swap the value of 2 variables without using 3rd variable
4. Write a program to calculate the sum of digits of a given integer no.
5. Write a program to compute the sum of the first and last digit of a given number.
6. Write a program in java which enter the number using Data Input Stream and check whether the entered number is even or odd.
7. Write an application that reads a string and determines whether it is a palindrome.
8. Write a program to enter a sentence form keyboard and also find all the words in that sentence with starting character as vowel.
9. Write a Program in java which creates the array of size 5; find the sum and average of the five numbers.
10. Create a java program that has three version of add method which can add two, three, and four integers.
11. Program illustrating Classes and Objects.
12. Program illustrating Method Overloading and Method Overriding.
13. Program illustrating concept of Interface.
14. Program illustrating use of Final and Super keyword.
15. Program that illustrates the Creation of simple package.
16. Program that illustrates the Accessing of a package.
17. Program that illustrates the Handling of predefined exceptions.
18. Program that illustrates the Handling of user defined exceptions.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Course Outcomes (COs):

At the end of this course students will be able to:

- Understand the fundamentals of Python programming.
- Make programs in Python using list.
- Make programs Python using dictionary.
- Make programs Python using string.
- Make programs Python using tuple.

LIST OF PRACTICALS

Instruction: At least 6 sections are to be implemented.

Section 1: Basic python program

- Python program to print Hello world!
- Python program to add two numbers
- Python program to find the square root
- Python program to calculate the area of a triangle
- Python program to swap two variables

Section 2: Python program on conversion

- Python program to convert kilometres to miles
- Python program to convert Celsius to Fahrenheit
- Python program to convert decimal to binary, octal and hexadecimal
- Python program to find ASCII value of character
- Python program to implement type conversion

Section 3: Basic mathematical program

- Python program to check Armstrong number
- Python program to check if a number is odd or even
- Python program to check leap year
- Python program to find the largest among three numbers
- Python program to check prime number

Section 4: Python program on list

- Python program to check if a list is empty
- Python program to access index of a list using for loop

Handwritten signatures and initials in blue ink at the bottom of the page.

- Python program to slice list
- Python program to concatenate two lists
- Python program to remove duplicate element from a list

Section 5: Python program on dictionary

- Python program to merge two dictionaries
- Python program to iterate over dictionary using for loop
- Python program to sort a dictionary by value
- Python program to delete an element from a dictionary
- Python program to check if a key is already present in a dictionary

Section 6: Python program on string

- Python program to check if given strings is palindrome or not
- Python program to capitalize the first character of a string
- Python program to compute all the Permutation of the String
- Python program to create a countdown timer
- Python program to count the number of occurrences of a character in string

Section 7: Python program on tuple

- Python program to find the size of a tuple
- Python program for adding a tuple to list and vice-versa
- Python program to sort a list of tuples in increasing order by the last element in each tuple
- Python program to assign frequency to tuples
- Python program to check if any list element is present in tuple

Section 8: Python program on Classes and Objects

- Create a class my class and add some element in it.
- Create a python program to access all elements of a given class
- Create a python program to show OOPs concept
- Create a python program to delete an object in python
- Create a class named Person, use the init() function to assign values for name and age

Section 9 : Python program on files

- Create a python program to make a file
- Create a python program to open and close a given file.
- Create a python program to read and write in file
- Create a python program for copying, moving, and renaming files
- Create a python program for deleting files in python

AS A J K CD AS

Section 10: Section 8: Python program on patterns

- Program to print full pyramid using *
- Pascal's triangle pattern using numbers
- Numbered Diamond pattern
- Square pattern in python
- Simple Number triangle pattern

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

AK