KOLHAN UNIVERSITY

Chaibasa, Jharkhand, India

Syllabus for
Four Year Undergraduate Programme (FYUGP)
of

Bachelor of Computer Application (BCA)

Semester - 4

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 – IV) for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)	03 – 03
2	MJ–6 (Theory): Java Programming Language – II	04 – 05
3	MJ–7 (Theory): Operating System – I	06 – 07
4	MJ–8: Software Engineering	08 – 09
5	MJ (Practical–4): Java Programming – II and Operating System – I Lab	10 – 12

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

Sem.	Paper Code	Paper Title	L–T–P	Credits	Contact Hours
IV	AEC–3	Language and Communication Skills *(To be selected by the students from the list of available options)*		2	
	VAC–2	Value Added Course – 2 *(To be selected by the students from the list of available options)*		2	
	MN–2B	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–6 (Theory)	Java Programming Language – II	3–0–0	3	45
	MJ–7 (Theory)	Operating System – I	3–0–0	3	45
	MJ–8	Software Engineering	3–1–0	4	60
	MJ (Practical–4)	Java Programming – II and Operating System – I 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)

MJ–6 (Theory): Java Programming Language-II

3 Credits | 45 Minimum Class Hours | Semester IV

Objectives:

The Java Programming Language course is designed to provide students with a comprehensive understanding of the Java programming language and its application in software development. The course aims to equip students with the necessary skills to design, implement, and debug Java programs. The main objectives of the course are as follows–

- Understand the concept of multithreading and its advantages and learn how to create and manage threads in Java.
- Explore different file handling techniques, such as byte and character streams.
- Explore the Java Collection Framework and its key interfaces and classes.
- Explore Java's Swing and JavaFX libraries for GUI development.
- Learn how to establish database connections and manage connections.

Learning Outcomes:

By the end of the course, students will be able to:

- Create and manage concurrent threads in Java applications.
- Perform read and write operations on text and binary files.
- Apply collection framework concepts to solve real-world problems.
- Develop interactive applications with responsive user interfaces.
- Develop robust database-driven Java applications.

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+N/A=75

Unit	Topic	Minimum Class Hours
I	Multithreading and Exception Handling	10
II	Managing Input/ Output Files in Java, Java Collection Framework	10
III	Event and GUI Programming	15
IV	JDBC	10
	Total	**45**

Detailed Syllabus

Unit I: Multithreading and Exception Handling **(10 Hours)**

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, Throwing Our Own Exceptions.

Unit II: Managing Input/ Output Files, Java Collection Framework (10 Hours)

Managing Input/ Output Files: Introduction, 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 III: Event and GUI programming (15 Hours)

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

AWT (Abstract Windows Toolkit) and Swings: Architecture, Component, Container, Panel, Window, Frame, Canvas, Components of AWT (Button, Label, Checkbox, CheckboxGroup, Choice, List, TextField etc.) and Swing (JLabel, JButton, JCheckBox, JRadioButton, JList, JComboBox, JTextField, JTextArea, JPanel, JFrame etc.), Working with Graphics, Working with Color and Font, Adding and Removing Controls, Responding To Controls, Layout Managers (Flow Layout, Grid Layout, Card Layout, Border Layout), Handling Events.

Unit IV: JDBC (10 Hours)

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–7 (Theory): Operating System – I

3 Credits | 45 Minimum Class Hours | Semester IV

Objectives:

The objective of this course is to introduce the students to a layer of software called the Operating Systems, whose job is to manage all the devices of a computer system and provide user programs with a simple interface to the hardware. This paper will familiarize the students with the concepts of process management, process synchronization, and the potential problem of deadlocks.

Learning Outcomes:

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

- Understand the basic working process of an operating system.
- Understand the importance of process and scheduling.
- Understand the issues in deadlock.

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+N/A=75

Unit	Topic	Minimum Class Hours
I	Overview	15
II	Process Management	20
III	Process Synchronization	10
	Total	**45**

Detailed Syllabus

Unit I: Overview **(15 Hours)**

Introduction: Definition of Operating Systems, Functions of Operating Systems, Generations of Operating System, Types of Operating System {Batch Operating System, Mainframe Operating System, Desktop Operating System, Time Sharing (or, Multi–tasking) Operating System, Multi–processing Operating System, Real–Time Operating System, Distributed Operating System, Clustered Operating System, Network Operating System, Mobile Operating System, Embedded Operating Systems}.

Operating System Structures: Operating System Components (Kernel, Hardware Device Drivers, Disk Access and File Systems, Security, User Interface), Operating System Services, System Calls, Application Programming Interface (API), User Mode and Kernel Mode, Types of System Calls, System Programs, Operating System Design and Implementation, Operating System Structure.

Unit II: Process Management **(20 Hours)**

Processes: Process Concept (The Process, Process State, Process Control Block), Process Scheduling (Scheduling Queues, CPU Scheduling, Context Switch), Operations on Processes (Process Creation, Process Termination), Process Termination in Android Operating System, Interprocess Communication (Independent Process, Co–operative Process, IPC in Shared–Memory Systems, IPC in Message–Passing Systems).

Multithreaded Programming: Introduction to thread (Components of Thread, Single–Threaded Process, Multithreaded Process), Differences between Process and Thread, Types of Thread (User Thread, Kernel Thread), Multithreading Models (Many–to–One, One–to–One, Many–to–Many), Advantages and Disadvantages of Thread.

CPU Scheduling: Basic Concepts (CPU–I/O Burst Cycle, CPU Scheduler, Preemptive and Non-preemptive Scheduling, Dispatcher), Scheduling Criteria, Scheduling Algorithms (First–Come, First–Served Scheduling, Shortest–Job–First Scheduling, Priority Scheduling, Round–Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling).

Unit III: Process Synchronization **(10 Hours)**

Overview: Introduction to Process Synchronization, Race Condition, The Critical–Section Problem, Peterson's Solution.

Deadlocks: Definition of a Deadlock, Conditions for Deadlock, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance (Resource–Allocation–Graph Algorithm, Banker's Algorithm), Deadlock Detection, Recovery from Deadlock (Process Termination, Resource Preemption).

Recommended Books:

- Abraham Silberschatz, Peter B. Galvin, Greg Gagne; **Operating System Concepts** (Tenth Edition); New York: John Wiley and Sons, 2018
- Andrew S. Tanenbaum; **Modern Operating Systems** (Third Edition); New Delhi: Prentice-Hall India, 2002
- William Stallings; **Operating Systems** (Fourth Edition), New Delhi: Prentice-Hall India, 2003

Further readings:

- Harvey M. Deitel; **Operating Systems** (Second Edition); New Delhi: Pearson Education
- Pramod Chandra P. Bhatt; **An Introduction to Operating Systems Concept**; New Delhi: Prentice-Hall India
- Maurice J. Bach; **The Design of the Unix Operating System**; New Delhi: Prentice-Hall India, 1992
- Brian W. Kernighan, Rob Pike; **The Unix Programming Environment**, Pearson Education, 1984

MJ–8: Software Engineering

4 Credits | 60 Minimum Contact Hours | Semester IV

Objectives:

This course helps students to understand the software development process and design. It also helps the students to understand about the different stages of software development, various process models and software engineering principles.

The main objectives of the course are as follows–

- To provide students an in depth understanding of software engineering principles.
- To prepare the students to develop the skills necessary to handle software projects.
- To make the students aware of the importance of software engineering principles in designing software projects.
- To make students familiar with cost estimation and testing measurement in software development process.

Course Outcomes:

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

- Understand the importance of the stages in the software lifecycle.
- Understand the various process models.
- To design software by applying the software engineering principles.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	N/A	3	N/A	4	N/A	20+5=25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=75

Unit	Topic	Minimum Class Hours
I	Software Engineering Concepts, Process Model	14
II	Requirements Analysis and Specification	14
III	Software Design	12
IV	Coding and Testing of Software	12
V	Software Maintenance	8
	Total	**60**

Detailed Syllabus

Unit I: Software Engineering Concepts **(14 Hours)**

Introduction**:** Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Quality Attributes.

SDLC or Software Development Process Models: Classical Waterfall Model, Iterative Waterfall Models, Prototype Model, RAD Model, Agile Development Models, Spiral Model.

Software Project Management**:** Size Estimation- Line of Code (LOC) and Function Point (FP) Metric, Cost Estimation-Delphi and Basic COCOMO Model.

Unit II: Requirements Analysis and Specification (14 Hours)

Software Requirements Specification (SRS): SRS Documents, their Characteristics and Organization.

Requirement Engineering Process**:** Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study.

Function Oriented Modeling**:** Data Flow Diagrams (DFD), Entity Relationship Diagrams (ERD), Data Dictionaries, and Decision Tables.

Unit III: Software Design (12 Hours)

Classification, Software Design Approaches, Cohesion and Coupling, Function Oriented Software Design, Introduction to Object Oriented Design, Need of UI design, Design issues, The UI design Process.

Unit IV: Coding and Testing of Software (12 Hours)

Coding: Coding Standards and Guidelines, Software Documentation.

Testing: Unit Testing, Black Box Testing, White Box Testing, Debugging, Program Analysis Tools, System Testing.

Unit V: Software Maintenance (08 Hours)

Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models, Estimation of Maintenance Cost.

Recommended Books:

- Rajib Mall; **Fundamentals of Software Engineering**; Prentice Hall of India, New Delhi
- Ian Sommerville; **Software Engineering**; Addison Wesley

Further Readings:

- Richard Fairley; **Software Engineering Concepts**; Tata McGraw Hill, New Delhi
- Pankaj Jalote; **Software Engineering**; Wiley

MJ (Practical-4): Java Programming – II and Operating System – I Lab

2 Credits | 60 Minimum Class Hours | Semester IV

Objectives:

The main objectives of the course are as follows–

- Understand the concept of multithreading and its advantages and learn how to create and manage threads in Java.
- Explore different file handling techniques, such as byte and character streams.
- Explore the Java Collection Framework and its key interfaces and classes.
- Explore Java's Swing and JavaFX libraries for GUI development.
- Learn how to establish database connections and manage connections.
- To make students familiar with the Microsoft Windows and Linux command–line environment. This course serves as a platform for the subsequent labs related to Process Management, Process Scheduling, etc.

Learning Outcomes:

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

- Create and manage concurrent threads in Java applications.
- Perform read and write operations on text and binary files.
- Apply collection framework concepts to solve real-world problems.
- Develop interactive applications with responsive user interfaces.
- Develop robust database-driven Java applications.
- Apply Microsoft Windows and Unix/Linux operating system basic commands.
- Understand different Commands for process management in Microsoft Windows and Linux Operating System.

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	N/A+50=50

Marks Distribution of End Semester Practical Examination

● Experiments – 30 Marks ● Viva-Voce – 10 Marks ● Practical File – 10 Marks

Experiment List

Group – 'A': Java Programming – II

Unit I: Multithreading and Exception Handling

1.	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.
2.	Write a Java program to use the try, catch and finally block to handle the exception.
3.	Write a program to illustrate the throws keyword in Java.
4.	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 II: Managing Input/ Output Files, Java Collection Framework	
5.	Write a java program to copy the contents from one file to another file.
6.	Write a java program to read the student data from user and store it in the binary file. Also read the file and display all the records.
7.	Write a program to implement Vector class and its methods to store N numbers into a vector and display its sum and average.
8.	Write a program to add, retrieve and remove the element from the ArrayList.
9.	Write a program to implement LinkedList and perform different operations on it.
Unit III: Event and GUI programming	
10.	Display various graphics object such as rectangle, square, oval, circle, lines etc. with different colours on screen.
11.	Write a Java program to create a simple calculator with basic operations such as +, -, /, * using java swing elements. Use appropriate layout.
12.	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.
Unit IV: JDBC	
13.	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.

Group – 'B': Operating System – I

Unit I: Basic Microsoft Windows and Linux Commands	
1.	Briefly explain the following basic Microsoft Windows commands with examples: logoff, shutdown, tasklist, taskkill, chkdsk, sfc /scannow, format, diskpart, ver, systeminfo, dir, cd, copy, del, ren, ipconfig, ping, tracert, netsh wlan show profile name=WiFi_SSID key=clear, cls, help, powercfg, clip, color, compact, erase, doskey, driverquery, exit, find, and hostname.
2.	Briefly explain the following basic Linux commands with examples: man, history, pwd, who, finger, passwd, exit, logout, shutdown, mkdir, cd, ls, cat, cp, cmp, mv, paste, rm, rmdir, find, more, head, tail, echo, sed, grep, awk, date, time, cal, diff, file, sort, chmod, chown, du, and compress.
Unit II: Different Commands for Process Management in Linux OS/Microsoft Windows OS	
1.	Launch a process (or, program or, application) in the foreground from terminal/command prompt. After launching the process, stop it during the execution.
2.	Display the list of running in foreground, running in background, force stopped, and pending processes.
3.	Resume the force stopped and pending processes while keeping them running in the background.
4.	Resume the force stopped and pending processes while keeping them running in the foreground.
5.	Launch a process in the background directly.
6.	Launch a process in the background directly without getting impacted by the closing of the terminal/command prompt.
7.	Display a dynamic real–time table of processes of your Linux/Windows operating system.
8.	How do we terminate a running process on our Linux/Windows operating system?
9.	Report the used and available space in the primary memory storage in human–friendly units like megabytes or kilobytes.
10.	Report the used and available space in the secondary memory storage in human–friendly units like megabytes or kilobytes.
11.	How can you prioritize a process as per your requirement?

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