

4. Mathematical Foundation of Computer Science, Y.N.Singh, New Age International		
BCS403 Object Oriented Programming with Java		
Course Outcome (CO)		Bloom's Knowledge Level (KL)
At the end of course , the student will be able to understand		
CO 1	Develop the object-oriented programming concepts using Java	K ₃ , K ₄
CO 2	Implement exception handling, file handling, and multi-threading in Java	K ₂ , K ₄
CO 3	Apply new java features to build java programs.	K ₃
CO 4	Analyse java programs with Collection Framework	K ₄
CO 5	Test web and RESTful Web Services with Spring Boot using Spring Framework concepts	K ₅
DETAILED SYLLABUS		3-1-0
Unit	Topic	Proposed Lecture
I	Introduction: Why Java, History of Java, JVM, JRE, Java Environment, Java Source File Structure, and Compilation. Fundamental, Programming Structures in Java: Defining Classes in Java, Constructors, Methods, Access Specifies, Static Members, Final Members, Comments, Data types, Variables, Operators, Control Flow, Arrays & String. Object Oriented Programming: Class, Object, Inheritance Super Class, Sub Class, Overriding, Overloading, Encapsulation, Polymorphism, Abstraction, Interfaces, and Abstract Class. Packages: Defining Package, CLASSPATH Setting for Packages, Making JAR Files for Library Packages, Import and Static Import Naming Convention For Packages	08
II	Exception Handling: The Idea behind Exception, Exceptions & Errors, Types of Exception, Control Flow in Exceptions, JVM Reaction to Exceptions, Use of try, catch, finally, throw, throws in Exception Handling, In-built and User Defined Exceptions, Checked and Un-Checked Exceptions. Input /Output Basics: Byte Streams and Character Streams, Reading and Writing File in Java. Multithreading: Thread, Thread Life Cycle, Creating Threads, Thread Priorities, Synchronizing Threads, Inter-thread Communication.	08
III	Java New Features: Functional Interfaces, Lambda Expression, Method References, Stream API, Default Methods, Static Method, Base64 Encode and Decode, ForEach Method, Try-with-resources, Type Annotations, Repeating Annotations, Java Module System, Diamond Syntax with	08

	Inner Anonymous Class, Local Variable Type Inference, Switch Expressions, Yield Keyword, Text Blocks, Records, Sealed Classes	
IV	Java Collections Framework: Collection in Java, Collection Framework in Java, Hierarchy of Collection Framework, Iterator Interface, Collection Interface, List Interface, ArrayList, LinkedList, Vector, Stack, Queue Interface, Set Interface, HashSet, LinkedHashSet, SortedSet Interface, TreeSet, Map Interface, HashMap Class, LinkedHashMap Class, TreeMap Class, Hashtable Class, Sorting, Comparable Interface, Comparator Interface, Properties Class in Java.	08
V	Spring Framework: Spring Core Basics-Spring Dependency Injection concepts, Spring Inversion of Control, AOP, Bean Scopes- Singleton, Prototype, Request, Session, Application, Web Socket, Auto wiring, Annotations, Life Cycle Call backs, Bean Configuration styles Spring Boot: Spring Boot Build Systems, Spring Boot Code Structure, Spring Boot Runners, Logger, BUILDING RESTFUL WEB SERVICES, Rest Controller, Request Mapping, Request Body, Path Variable, Request Parameter, GET, POST, PUT, DELETE APIs, Build Web Applications	08
Text Books <ol style="list-style-type: none"> 1. Herbert Schildt, "Java The complete reference", McGraw Hill Education 2. Craig Walls, "Spring Boot in Action" Manning Publication 1. Steven Holzner, "Java Black Book", Dreamtech. 2. Balagurusamy E, "Programming in Java", McGraw Hill 3. Java: A Beginner's Guide by Herbert Schildt, Oracle Press 4. Greg L. Turnquist "Learning Spring Boot 2.0 - Second Edition", Packt Publication 5. AJ Henley Jr (Author), Dave Wolf, "Introduction to Java Spring Boot: Learning by Coding", Independently Published 		

BCS451- Operating System Lab

List of Experiments (Indicative & not limited to)

1. Study of hardware and software requirements of different operating systems (UNIX,LINUX,WINDOWS XP, WINDOWS7/8)
2. Execute various UNIX system calls for
 - i. Process management
 - ii. File management
 - iii. Input/output Systems calls
3. Implement CPU Scheduling Policies:
 - i. SJF
 - ii. Priority
 - iii. FCFS
 - iv. Multi-level Queue
4. Implement file storage allocation technique:
 - i. Contiguous(using array)

- ii. Linked –list(using linked-list)
 - iii. Indirect allocation (indexing)
- 5. Implementation of contiguous allocation techniques:
 - i. Worst-Fit
 - ii. Best- Fit
 - iii. First- Fit
- 6. Calculation of external and internal fragmentation
 - i. Free space list of blocks from system
 - ii. List process file from the system
- 7. Implementation of compaction for the continually changing memory layout and calculate total movement of data
- 8. Implementation of resource allocation graph RAG)
- 9. Implementation of Banker`s algorithm
- 10. Conversion of resource allocation graph (RAG) to wait for graph (WFG) for each type of method used for storing graph.
- 11. Implement the solution for Bounded Buffer (producer-consumer)problem using inter process communication techniques-Semaphores
- 12. Implement the solutions for Readers-Writers problem using inter process communication technique - Semaphore

BCS452- Object Oriented Programming with Java

List of Experiments (Indicative & not limited to)

1. Use Java compiler and eclipse platform to write and execute java program.
2. Creating simple java programs using command line arguments
3. Understand OOP concepts and basics of Java programming.
4. Create Java programs using inheritance and polymorphism.
5. Implement error-handling techniques using exception handling and multithreading.
6. Create java program with the use of java packages.
7. Construct java program using Java I/O package.
8. Create industry oriented application using Spring Framework.
9. Test RESTful web services using Spring Boot.
10. Test Frontend web application with Spring Boot

BCS453- Cyber Security Workshop

List of Experiments (Indicative & not limited to)

Module 1: Packet Analysis using Wire shark

1. Basic Packet Inspection: Capture network traffic using Wire shark and analyze basic protocols like HTTP, DNS, and SMTP to understand how data is transmitted and received.