BCSC1002: Object-Oriented Programming (New)

OBJECTIVE

This course introduces the Object-Oriented programming paradigm to students. It also teaches a student how to think objectively and model a Java program for solving real-world problems.

CREDITS: 3 **L-T-P:3-0-0**

Module No.	Content	Teaching Hours
I	**Object-Oriented Programming:** Features of Object-Oriented Programming, Introduction to Object-Oriented Java Programming. **Understanding Java Technology & Environment:** Understanding the compilation process of the JVM, JVM vs JDK vs JRE, Key Features of Java, Structure of a simple Java program. **Working with Java Primitive Data Types:** Strongly Typed nature of Java, Primitive Data Types in Java, The new 'var' keyword, Scope of a variable. **Accepting User Input in Java Programs:** using the Scanner class, using command line arguments. **Programming Constructs:** Sequence, Selection, Iteration & Transfer Statements, For-Each Loop. **Working with Java Arrays**: Declaring and Initializing One-Dimensional and Two-Dimensional Arrays in Java, Introduction to java.util.Arrays class. **The String API:** String Data Type, commonly used methods from the String API, StringTokenizer, StringBuilder & StringBuffer. **Creating and Using Methods:** Signature of a method, Types of Methods, Overloading methods in a class, Static and Non-Static Methods.	14 hours
II	**Describing and Using Objects & Classes:** Declare the structure of a Java class, declaring members of a class (fields and methods), declaring and using Java Objects, lifecycle of an Object (creation, assignment, dereferencing and garbage collection), Constructors of a class, Overloading Constructors, Constructor chaining using 'this' and 'super' keyword. **Using Java Packages:** create and import Java packages and static imports, abstracting program logic to packages, creating executable main class, running the executable class inside a package. **Applying Encapsulation**: Using access modifiers with/in a class, principles of encapsulation. **Programming Abstractly Through Interfaces:** create and implement Interfaces for programs, private and default methods in Interfaces, declaring Abstract Classes, Constructors in Abstract Classes. Marker Interface, Functional Interfaces, Lambda Expressions in Java. **Reusing Implementations using Inheritance:** Declaring Subclasses and Superclasses, extend Abstract Classes, implementing Interfaces, exploring polymorphic behaviour by overriding methods, Object Types vs Reference Types, differentiate overloading, overriding and hiding. **Exception Handling:** Exception Hierarchy, Need of Exception Handling, Checked Exceptions, Unchecked Exceptions and Errors, Try-Catch Blocks, Finally, Throw & Throws Keywords, creating and handling Custom Exceptions.	14 hours
III	**Threads in Java:** Life Cycle of a Thread, Creating threads using Runnable and Thread, 'sleep()', Thread Priorities. **Using Wrapper Classes:** Wrapper Classes in Java, Boxing-Unboxing-Autoboxing-AutoUnboxing. **Generics & Collections:** Creating Generic classes, Generic Methods, Diamond Notation, Wildcards, Type Erasure, Collection Hierarchy, Base Interfaces, Lists, Sets and Maps. **The Stream API:** Introduction to the Stream API, using lambda expressions in Streams. **Regular Expressions:** Pattern and Matcher Class. **JDBC:** JDBC Drivers, Connecting to a MySQL Database, DriverManager, Connection Interface, Statement Interface, ResultSet Interface, PreparedStatements.	14 hours

Text Books:

- Herbert Schildt (2019), "The Complete Reference, Java Eleventh Edition", Oracle Press.

Reference Books:

- Cay S Hosrtmann (2018), "Core Java Volume I—Fundamentals, Eleventh Edition", Pearson
- Rogers Cadenhead (2020), "Sams Teach Yourself Java in 21 Days (Covers Java 11/12), 8th Edition", Pearson