**JAVA Syllabus:**

**Variables and Data Types**:

* Primitive data types (int, float, double, char, boolean, etc.)
* Reference types (String, arrays, classes)

 **Operators and Expressions**:

* Arithmetic operators
* Relational operators
* Logical operators
* Assignment operators
* Increment and decrement operators

 **Control Flow Statements**:

* Conditional statements (if, if-else, switch)
* Looping statements (for, while, do-while)
* Break and continue statements

 **Methods**:

* Defining methods
* Method overloading
* Method parameters and return types
* Static methods vs instance methods

 **Object-Oriented Programming (OOP) Concepts**:

* Classes and objects
* Constructors
* Inheritance
* Method overriding
* Polymorphism
* Encapsulation
* Abstraction

 **Exception Handling**:

* Try-catch block
* Finally block
* Throw and throws keywords
* Custom exceptions

 **Collections Framework**:

* Lists (ArrayList, LinkedList)
* Sets (HashSet, TreeSet)
* Maps (HashMap, TreeMap)
* Iterating over collections

 **File I/O**:

* Reading from and writing to files
* BufferedReader and BufferedWriter
* FileInputStream and FileOutputStream

 **Java Standard Library (java.lang, java.util, java.io, etc.)**:

* String manipulation (String, StringBuilder, StringBuffer)
* Date and time (LocalDate, LocalTime, LocalDateTime)
* Utility classes (Math, Arrays, Collections)

 **Multithreading and Concurrency**:

* Creating threads (Thread class, Runnable interface)
* Synchronization
* Executors framework

 **Generics**:

* Generic classes and methods
* Bounded types
* Wildcards

 **Lambda Expressions and Stream API** (Java 8 and above):

* Functional interfaces
* Lambda expressions
* Stream operations (filter, map, reduce, etc.)

 **Annotations and Reflection**:

* Built-in annotations (Override, Deprecated, SuppressWarnings)
* Custom annotations
* Reflection API

 **Basic GUI Programming (Optional)**:

* Swing or JavaFX

 **Build Tools and Dependency Management**:

* Maven or Gradle