**1.** [**Basics of Java**](Basics%20of%20Java.docx)

* Introduction to Java and its features.
* Setting up the Java environment (JDK).
* Writing a "HelloWorld" program and understanding Java structure.
* Importance of comments in code.

**2.** [**Variables, Data Types, and Operators**](Variables,%20Data%20Types,%20and%20Operators.docx)

* Understanding variables, constants, and data types (primitive/non-primitive).
* Type casting and operator usage (arithmetic, relational, logical, etc.).

**3. [Control Statements](Control%20Statements.docx)**

* Conditional statements: if, else if, else, switch.
* Loops: for, while, do-while, enhanced for.
* Special statements: break, continue, return.

**4.** [**Arrays**](Arrays.docx)

* One-dimensional and multi-dimensional arrays.
* Array operations like traversing, sorting, and searching.
* Using the Arrays utility class.

**5.** [**Functions and Methods**](Functions%20and%20Methods.docx)

* Defining and calling methods with parameters.
* Method overloading and the return keyword.
* Handling variable arguments (varargs).

**6.** [**Object-Oriented Programming (OOP)**](Object-Oriented%20Programming%20(OOPS).docx)

* Core principles: Encapsulation, Abstraction, Inheritance, Polymorphism.
* Working with classes, objects, constructors, and static members.
* Nested/inner classes and the this keyword.

**7.** [**Inheritance and Interfaces**](Inheritance%20and%20Interfaces.docx)

* Types of inheritance, method overriding, and the super keyword.
* Abstract classes and interfaces, including default and static methods.

**8.** [**Packages and Access Modifiers**](Packages%20and%20Access%20Modifiers.docx)

* Organizing code with packages.
* Access levels: public, private, protected, default.
* Using import and static imports.

**9.** [**Exception Handling**](Exception%20Handling.docx)

* Handling checked and unchecked exceptions.
* try, catch, finally, and custom exceptions.
* Using throw and throws.

**10.** [**Strings and String Handling**](Strings%20and%20String%20Handling.docx)

* Manipulating strings with String, StringBuilder, and StringBuffer.
* String operations: concatenation, comparison, splitting, and regex.

**11.** [**Collections Framework**](Collections%20Framework.docx)

* Data structures: List, Set, Map, and their implementations (e.g., ArrayList, HashMap).
* Iterators, enhanced for loops, and generics.

**12.** [**File Handling**](File%20Handling.docx)

* Reading/writing files using streams and classes like BufferedReader.
* Understanding serialization and deserialization.

**13.** [**Multithreading and Concurrency**](Multithreading%20and%20Concurrency.docx)

* Creating threads using Thread and Runnable.
* Thread lifecycle, synchronization, and inter-thread communication.
* Thread pools (ExecutorService).

**14.** [**Java 8 Features (and Beyond)**](Java%208%20Features%20(and%20Beyond).docx)

* Lambda expressions, functional interfaces, and the Stream API.
* Optional class and new Date/Time API.
* Default and static methods in interfaces.

**15.** [**Advanced Topics**](Advanced%20Topics.docx)

* Annotations, Reflection API, and Networking (Sockets, HTTP).
* JDBC for database connectivity.
* GUI development with JavaFX or Swing.
* Design patterns for effective programming.

**16.** [**Frameworks and Tools**](Frameworks%20and%20Tools.docx)

* Explore frameworks like Spring (Spring Boot, Spring MVC) and Hibernate (ORM).
* Use build tools like Maven/Gradle and testing tools like JUnit/Mockito.

**Tips for Success:**

* Practice coding exercises and build mini-projects.
* Refer to Java documentation and engage in developer communities.
* Work on real-world applications to reinforce concepts.