Milestone 1: Introduction and Basics (0:00:00 - 0:48:11)

Topics Covered: JDK Setup, First Code, Variables, Data Types.

Project: Simple Calculator

- **Description**: Build a console-based calculator that performs basic operations: addition, subtraction, multiplication, and division.

Success Criteria:

- Correct handling of user input for two numbers and an operator.
- Proper error handling (e.g., division by zero).
- Use of variables and correct data types for calculations.

Milestone 2: Operators and Conditional Statements (0:48:11 - 1:46:47)

Topics Covered: Operators (Assignment, Relational, Logical), If-Else, Ternary, Switch.

Project: Grading System

- **Description**: Write a program to input a student's marks and output their grade using if-else and switch statements.

Success Criteria:

- Takes input for a student's marks.
- Uses conditional statements to assign a grade (e.g., A for 90+, B for 80-89, etc.).
- Implement both if-else and switch statements.

Milestone 3: Loops (1:46:47 - 2:50:06)

Topics Covered: While Loop, Do-While Loop, For Loop.

Project: Number Guessing Game

- **Description**: Implement a number guessing game where the computer generates a random number, and the user guesses until they find the correct number.

Success Criteria:

- Use a loop to allow continuous guesses.

- Provide feedback if the guess is too high or low.
- Properly handle input validation and loop termination.

Milestone 4: Classes, Objects, and Methods (2:50:06 - 3:48:12)

Topics Covered: Classes, Objects, Methods, Method Overloading.

Project: Bank Account System

- **Description**: Create a `BankAccount` class that allows users to deposit, withdraw, and check their balance.

Success Criteria:

- Define a `BankAccount` class with fields for balance, account number, etc.
- Implement methods for deposit, withdrawal, and checking balance.
- Use method overloading for different types of deposits (e.g., by amount or by check).

Milestone 5: Arrays and Strings (3:48:12 - 4:48:43)

Topics Covered: Arrays, Multi-dimensional Arrays, Strings, StringBuffer vs StringBuilder.

Project: Student Score Manager

- **Description**: Create a system to manage and manipulate student scores using arrays.

Success Criteria:

- Store student names and scores in arrays.
- Implement methods to add, update, and display student scores.
- Perform sorting and searching on the student array.

Milestone 6: Constructors and Inheritance (5:37:36 - 6:36:16)

Topics Covered: Constructors, Inheritance, Method Overriding.

Project: Library System

- **Description**: Create a `Book` class and an `EBook` subclass with additional fields like file size.

Success Criteria:

- Use constructors for initializing `Book` and `EBook`.
- Implement method overriding in `EBook` to show details.
- Test inheritance by creating objects of both classes and calling overridden methods.

Milestone 7: Polymorphism and Interfaces (7:04:42 - 8:27:11)

Topics Covered: Polymorphism, Interfaces, Abstract Classes.

Project: Shape Drawing Application

- **Description**: Create a set of classes for different shapes (e.g., Circle, Rectangle) that implement a `Drawable` interface.

Success Criteria:

- Implement an interface `Drawable` with a `draw()` method.
- Create multiple shape classes that implement this interface.
- Demonstrate polymorphism by handling shapes using the `Drawable` interface.

Milestone 8: Exception Handling (9:10:41 - 10:07:17)

Topics Covered: Try-Catch, Exception Hierarchy, Custom Exceptions.

Project: Custom Calculator with Error Handling

- **Description**: Modify your previous calculator project to handle exceptions (e.g., division by zero) and throw custom exceptions for invalid input.

Success Criteria:

- Use try-catch blocks to handle errors.
- Implement a custom exception for invalid operator input.
- Ensure the program doesn’t crash on invalid input and gracefully handles errors.

Milestone 9: Multithreading (10:15:25 - 11:00:15)

Topics Covered: Threads, Runnable, Race Conditions.

Project: Multithreaded Counter

- **Description**: Create a program where multiple threads increment different counters.

Success Criteria:

- Use the `Thread` class and the `Runnable` interface to implement multiple threads.
- Synchronize access to shared resources to avoid race conditions.
- Provide control over thread execution (start/stop threads).

Milestone 10: Collections and Stream API (11:03:45 - 12:14:09)

Topics Covered: ArrayList, Set, Map, Stream API.

Project: Employee Management System

- **Description**: Build a system to store employee records using a list. Implement functionality to add, update, remove, and display employees.

Success Criteria:

- Use collections like `ArrayList` to store employee records.
- Implement sorting, filtering, and searching using the Stream API.
- Demonstrate the use of lambda expressions in stream operations.