1. **Boolean Expressions and Logical Operators**
    - Introduction to boolean (true/false) expressions.
    - Use of relational operators (>, <, ==, !=, etc.) and logical operators (&&, ||, !).
    - Short-circuited evaluation of logical expressions (where `&&` and `||` may skip evaluating the right-hand side if the left-hand side already determines the result).
2. **Conditional Statements (if, if-else)**
    - How to direct the flow of a program using conditions.
    - Proper use of braces to create blocks and the significance of indentation.
    - Nested if statements for more complex decision-making.
3. **Comparing Different Types of Data**
    - The nuances of comparing floating-point numbers (and why direct equality checks can be problematic).
    - Comparing characters using their underlying numeric (Unicode) values.
    - Comparing strings using `equals` (for equality) and `compareTo` (for lexicographic ordering).
    - Distinguishing between equality of object references (`==`) and object content (`equals`).
4. **Repetition (while Loops)**
    - The basic structure and logic of the `while` statement for repeated execution.
    - The concept of sentinel values to terminate input-driven loops.
    - Ensuring loops terminate properly to avoid infinite loops.
    - Nested while loops for multidimensional repetition scenarios.
5. **Iterators**
    - The idea of an iterator that processes a collection of items one by one.
    - How the `Scanner` class implements the `Iterator` interface (hasNext, next, etc.).
6. **The ArrayList Class**
    - An introduction to `ArrayList<E>` for managing a dynamic list of objects.
    - Key methods like `add`, `remove`, `get`, `size`, and the concept of indices starting at 0.
    - The ability to grow and shrink an `ArrayList` as needed, in contrast to a fixed-size array.

These topics lay the groundwork for making decisions (conditionals), handling repetition (loops), and working with collections (iterators and `ArrayList`), all of which are crucial for structuring more complex programs.