Introduction to Java

1 Basic Data Types

In Java, there are eight primitive data types. They are:

- **byte**: 8-bit signed integer.
- **short**: 16-bit signed integer.
- **int**: 32-bit signed integer.
- **long**: 64-bit signed integer.
- **float**: Single-precision 32-bit IEEE 754 floating point.
- **double**: Double-precision 64-bit IEEE 754 floating point.
- **char**: Single 16-bit Unicode character.
- **boolean**: Represents one of two values: true or false.

Each of these data types has its own use cases, depending on the needs of the application.

2 Strings

A `String` in Java is a sequence of characters. It is an object that represents a collection of characters, and it is immutable, meaning that once a `String` is created, it cannot be changed.

```
String greeting = "Hello, World!";
```

To manipulate strings, Java provides a variety of methods in the `String` class. For example:

```
String name = "Alice";
String message = greeting + " My name is " + name;
System.out.println(message);
```

This code concatenates multiple strings and outputs: `Hello, World! My name is Alice`.

3 Syntax of Java

Java has a specific syntax that must be followed for the code to compile correctly. Here are some key aspects:

3.1 Comments

Comments are used to explain the code and are ignored by the compiler. There are two types of comments in Java:

- `Single-line comment:` Starts with `//` and continues to the end of the line.
- `Multi-line comment:` Enclosed between `/*` and `*/`.

3.2 Classes and Methods

Java is an object-oriented programming language, and everything is part of a class. Here's an example of a simple class:

```
public class MyClass {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

The `main` method is the entry point of any Java program.

3.3 Variables

Variables must be declared before they can be used. The syntax for declaring a variable is:

```
dataType variableName;
```

Example:

```
int count;
String name;
```

4 Conditional Statements

Conditional statements allow you to execute certain pieces of code based on whether a condition is true or false. The most common conditional statements in Java are `if`, `else if`, and `else`.

Here is an example:

```
int number = 10;

if (number > 0) {
    System.out.println("The number is positive.");
} else if (number < 0) {
    System.out.println("The number is negative.");
} else {
    System.out.println("The number is zero.");
}
```

In this example, the program checks whether the variable `number` is positive, negative, or zero and prints the corresponding message.

5 For Loops

A `for` loop is a control flow statement that allows code to be executed repeatedly based on a given condition. It is often used to iterate over arrays or collections.

The syntax of a `for` loop is as follows:

```
for (initialization; termination; increment) {
    // Code to be executed
}
```

Here's an example that prints numbers from 1 to 5:

```
for (int i = 1; i <= 5; i++) {
    System.out.println(i);
}
```

This loop initializes the variable `i` to 1, checks if `i` is less than or equal to 5, and increments `i` by 1 in each iteration, printing the current value of `i`.