Java Basics Course Study Materials

2. Basic Syntax and Structure

2.1 Java Syntax:

Java is a case-sensitive programming language. This means that System, system, and SYSTEM are considered different identifiers. Every Java application must have at least one class definition that contains the main method. This main method serves as the entry point for the application.

2.2 Class Declaration:

A class is a blueprint for creating objects. A typical class declaration includes the class keyword followed by the class name.

```
public class MyClass {
    // Class body
}
```

2.3 Main Method: The main method is the entry point of a Java application. It is defined as:

```
public static void main(String[] args) {
    // Code to be executed
}
```

- **public:** The method is accessible from anywhere.
- **static:** The method belongs to the class rather than an instance of the class.
- **void:** The method does not return any value.

- **String[] args:** The method accepts an array of String arguments.
-

Print Statement: The `System.out.println` method is used to print messages to the console.

```
System.out.println("Hello, World!");
```

2.4 Comments: Comments are used to explain the code and are ignored by the compiler.

- **Single-line comment:**

```
// This is a single-line comment
```

- **Multi-line comment:**

```
/* This is a
   multi-line comment */
```

- **Documentation comment:**

```
/**
 * This is a documentation comment
 */
```

2.5 Java Identifiers: Identifiers are names given to elements such as classes, methods, and variables. They must begin with a letter, underscore, or dollar sign and can be followed by letters, digits, underscores, or dollar signs.

- **Valid identifiers:** `myVariable`, `_myVariable`, `$myVariable`
- **Invalid identifiers:** `1stVariable`, `my-Variable`

2.6 Java Keywords: Keywords are reserved words in Java that have a predefined meaning and cannot be used as identifiers. Examples include class, public, static, void, int, etc.

2.7 Naming Conventions: Following naming conventions makes code more readable and maintainable.

- **Class names:** Start with an uppercase letter and use CamelCase. Examples: MyClass, HelloWorld.
- **Method names:** Start with a lowercase letter and use camelCase. Examples: myMethod, printMessage.
- **Variable names:** Start with a lowercase letter and use camelCase. Examples: myVariable, totalAmount.
-

Example: Simple Java Program

```
public class Main {
    public static void main(String[] args) {
        // Declaring a variable
        int number = 10;

        // Printing the variable value
        System.out.println("The number is: " + number);
    }
}
```