**KEYWORDS, IDENTIFIERS AND DATA TYPES**

Keywords are reserved words which have specific meaning in Java.

* We should not use keywords as identifiers or else we’ll get errors.

**Data Types:**

* **boolean (1 bit)**
* **byte (8 bits)**
* **short (16 bits)**
* **int (32 bits)**
* **long (64 bits)**
* **float (32 bits)**
* **double (64 bits)**
* **char (16 bits)**

These above are the **reserved keywords as data types**.

**Control Statements:**

* if: The if statement specifies a block of Java code to be executed if a condition is true.
* else: this keyword is used in conjunction with if to create conditional statements.
* for: keyword used to create a loop that iterates a specified number of times.
* while: keyword which is used to create a loop that continues to execute a block of statements if a specified condition is true.
* do-while: used to create a loop that executes a block of statements once, and then repeatedly executes the block as long as a specified condition is true.
* switch: used to implement multi-way branching based on the value of an expression.
* case: used within a switch statement to specify different possible values that a variable or expression might have.
* break: used to exit the loop prematurely, regardless of whether the loop condition still holds true.
* continue: statement skips the remaining code in the current iteration and moves directly to the next iteration.
* default: specify the code that should execute if none of the case constants match the value of the expression being evaluated.

These above are the **reserved keywords as Control Statements**.

**Access Modifiers:**

* public: used to declare that a class, method, or field is accessible from any other class.
* private: used to restrict access to classes, methods, and fields.
* protected: used to restrict access to members (fields, methods, and constructors) within the same package and by subclasses outside the package.

These above are the **reserved keywords as Access Modifiers**.

**OOPS:**

* new: an operator used to create new instances (objects) of classes.
* enum: data type used to define a collection of constants. (Specialized class interface)
* this: refers to the current instance of a class.
* super: used to refer to the superclass (parent class) of the current object instance.
* abstract: used to declare abstract classes and abstract methods.

These above are the **reserved keywords as OOP keywords**.

**Modifiers:**

* static: keyword that can be applied to fields, methods, blocks, and nested classes. It indicates that the member belongs to the class itself rather than to instances of the class.
* final: keyword is used to declare constants, methods, classes, and variables with specific behaviors. It signifies that once initialized or defined, the declared entity cannot be changed or overridden.
* synchronize: used to achieve thread safety by preventing multiple threads from concurrently executing a synchronized block of code or a synchronized method on the same object. It ensures that only one thread can execute the synchronized code block or method at a time, while other threads wait until the lock is released.
* transient: used to indicate that a member variable of a class should not be serialized when the object is converted into a stream of bytes (for example, when the object is written to a file or sent over a network).
* strictfp: used to restrict floating-point calculations to ensure consistent results across different platforms. It stands for "strict floating-point" and is primarily used with classes, methods, and interfaces.
* void: used to specify that a method does not return any value.
* return: used to exit from a method and optionally return a value to the calling code.
* native: used to indicate that a method is implemented in platform-dependent code (typically written in another programming language like C or C++) rather than in Java itself.
* assert: used primarily for debugging purposes to make assumptions about code behavior and validate those assumptions during program execution.
* const: define the constant value that cannot change during program execution.
* goto: it is used to alter the normal flow of control in a program.
* volatile: used to indicate that a variable's value may be changed unexpectedly by multiple threads. It ensures visibility of changes to variables across threads and prevents threads from caching variables when they are not supposed to.
* try: used as part of exception handling to define a block of code where exceptions may occur.
* catch: used as part of the exception handling mechanism to handle specific types of exceptions that may be thrown within a try block.
* finally: used in conjunction with a try block to ensure that certain code executes, regardless of whether an exception was thrown or not. It provides a mechanism for performing cleanup actions, such as closing resources like files or database connections, releasing locks, or finalizing operations that should be executed under all circumstances.
* throw: used to explicitly throw an exception. It allows you to create and throw custom exceptions or propagate built-in exceptions to indicate abnormal conditions or errors during program execution.
* throws: used in method declarations to indicate that a particular method may throw one or more exceptions during its execution.

These above are the **reserved keywords as Modifiers**.

String str = null

Default value of string: So, for any objects the default value is NULL.

**NOTE: All keywords should be in small cases**.

Anything as ‘final’ -> Can’t be extended further.

**IDENTIFIERS**

* Any name which is used to identify or refer uniquely that is an identifier.

**RULES FOR IDENTIFIER**

* Always identifier should have letters (English letters) and digits (any numbers).
* Always identifier name must start with letters and some symbol (\_, $) etc, but not with digits.
* We shouldn’t use any keywords as identifier.