Glossary

Control Structures and String Handling

Welcome! This alphabetized glossary contains many terms used in this module. Understanding these terms is essential when working in the industry, participating in user groups, and participating in other certificate programs.

Term	**Definition**
Abstract class	A class that cannot be instantiated and can have abstract methods.
Abstraction	The concept of hiding implementation details and exposing functionality.
Access modifiers	Keywords that control the visibility of methods and variables.
Annotation	A special type of metadata that provides information about code.
API (Application Programming Interface)	A set of functions and protocols for building and integrating applications.
Arguments	The actual values passed to a function or method.
Array	A fixed-size collection of elements of the same type.
ArrayList	A resizable array implementation in the java.util package.
break	A keyword that exits a loop or switch statement.
BufferedReader	A class used for efficient reading of text from input streams.
charAt	A method used to access a character at a specific index in a string.
class	A blueprint for creating objects in Java.
Comparable	An interface that allows objects to be sorted based on natural ordering.
Comparator	An interface used for defining custom sorting logic.
concat	A method that joins two strings together.
Concatenation	The process of combining two or more strings.
Constructor	A special method used to initialize objects.
continue	A keyword that skips the current iteration of a loop.
Deserialization	The process of converting a byte stream back into an object.
Encapsulation	The practice of keeping data private and providing controlled access.
enum	A special class representing a fixed set of constants.
equals	A method that checks if two strings have the same content.
Exception	An error that occurs during program execution.
File	A class representing file and directory paths in Java.
final	A keyword used to declare constants or prevent inheritance.
finally	A block of code that executes after try-catch, whether an exception occurs or not.
Garbage collection	The automatic process of reclaiming unused memory.
Generics	A feature enabling type-safe operations on collections and classes.
HashMap	A key-value pair collection in Java.
if-else	A conditional statement that executes different code based on conditions.
Immutable	A property of String, meaning it cannot be modified after creation.
import	A statement used to bring external classes or packages into a program.
Inheritance	The mechanism of acquiring properties from a parent class.
InputStream	A class used for reading byte streams.
Instance method	A method associated with an instance of a class, requiring an object to be invoked.
interface	A contract that a class can implement, defining method signatures.
java.io	A package that handles input and output operations.
java.lang	A built-in package that contains fundamental Java classes.
java.net	A package that provides networking capabilities.
java.sql	A package used for database connectivity.
java.time	A package introduced in Java 8 for modern date and time handling.

Term	Definition
java.util	A package that provides utility classes for data structures and algorithms.
JDBC (Java Database Connectivity)	An API for database interaction.
join	A method that combines elements of an array into a single string.
Lambda	A concise way to represent anonymous functions introduced in Java 8.
length	A method that returns the number of characters in a string.
Loop	A control structure that repeats a block of code.
Method	A block of code that performs a specific task in a class.
Method signature	The combination of a method name and parameter list that defines a method.
Module	A feature introduced in Java 9 for better dependency management.
new	A keyword used to create a new object explicitly.
Object	An instance of a class containing state and behavior.
Optional	A container object introduced in Java 8 to handle null values safely.
OutputStream	A class used for writing byte streams.
Overloading	The process of defining multiple methods with the same name but different parameters.
Package	A namespace that groups related classes together.
Parameters	The inputs provided to a function or method.
Polymorphism	The ability of different classes to be treated as instances of the same class.
Record	A compact class type introduced in Java 14 for immutable data storage.
Recursion	A technique where a method calls itself.
Reflection	A feature allowing inspection and modification of classes at runtime.
replace	A method that replaces a character or substring with another value.
REST (Representational State Transfer)	A web service architecture that uses HTTP requests for communication.
Return type	The data type of the value returned by a function or method.
Scope	The accessibility of a variable or method within a program.
Serialization	The process of converting an object into a byte stream.
split	A method that divides a string into parts based on a delimiter.
static	A keyword used to define class-level methods and variables.
Static method	A method that belongs to a class rather than an instance and can be called without creating an object.
Stream	A sequence of elements supporting functional-style operations.
String	A sequence of characters used to represent text in Java.
String literal	A way to create a string by enclosing text in double quotes.
substring	A method that extracts a part of a string.
switch	A control structure that allows multiple execution paths.
synchronized	A keyword ensuring that only one thread can access a block of code at a time.
this	A keyword that refers to the current object of a class.
Thread	A lightweight process enabling concurrent execution.
throw	A keyword used to manually trigger an exception.
throws	A declaration that a method may throw an exception.
toLowerCase	A method that converts all characters in a string to lowercase.
toUpperCase	A method that converts all characters in a string to uppercase.
trim	A method that removes whitespace from the beginning and end of a string.
try-catch	A mechanism to handle exceptions gracefully using try and catch blocks.
var	A keyword introduced in Java 10 for local variable type inference.
void	A return type indicating a method does not return a value.
volatile	A keyword ensuring that a variable's value is always read from the main memory.

Skills Network