

CPSC 319
Data Structures, Algorithms, and Their Applications
Winter 2024

Strings

- In Java, a String is a sequence of characters.
- Strings are widely used for representing text and are considered objects.
- Strings in Java are immutable, meaning their values cannot be changed once they are created.
- Operations on strings create new strings rather than modifying the existing ones.

Enumerations

- Enum Types, short for enumerations, are a special data type in Java.
- They provide a way to represent a fixed set of named constants.
- Enums are used to model a collection of related constants or options.
- They offer a cleaner and more readable alternative to using integers or strings for representing fixed values.

Wrappers

- Wrappers, short for wrapper classes, are classes in Java that encapsulate primitive data types.
- They provide a way to treat primitive types as objects.
- Java is primarily an object-oriented language, but it includes primitive data types (int, char, double, etc.) that are not objects.
- Wrappers bridge the gap by allowing these primitives to be used where objects are required.
- Each primitive type has a corresponding wrapper class.
- Examples: Integer for int, Double for double, Character for char, etc.
- Autoboxing is the automatic conversion of a primitive type to its corresponding wrapper class. Unboxing is the automatic conversion of a wrapper class to its primitive type.

Expressions and Operators

- Operators in Java are symbols that perform operations on variables and values.
- In Java, an expression is a combination of variables, operators, and method calls that produces a single value.

Control Flow

- Control flow refers to the order in which statements are executed in a program.
- It includes decision-making and looping structures for directing program execution.

Conditional Statements

- **if** Statement

Executes a block of code if a specified condition is true.

- **else** Statement

Provides an alternative block of code to execute when the if condition is false.

- **switch-case** Statement

Evaluates an expression against multiple possible case values.

Looping Statements

- **for** Loop
Iterates a block of code a specified number of times.
- **while** Loop
Repeats a block of code as long as a specified condition is true.
- **do-while** Loop
Similar to while loop but guarantees the block of code is executed at least once.
- **break** Statement
Terminates the loop prematurely.
- **continue** Statement
Skips the rest of the loop's code for the current iteration and moves to the next iteration.

Branching Statements

- **return** Statement

Exits a method and returns a value (if the method has a return type).

- **throw** Statement

Throws an exception manually, interrupting normal program flow.

Exception Handling

- **try-catch** Block

Used for handling exceptions to prevent program termination.

- **finally** Block

Executes regardless of whether an exception occurred or not.

Input and Output

- Java provides the **Scanner** class for handling various types of input.
- Instantiate a **Scanner** object associated with a specific input source (e.g., **System.in**).
- Java uses **System.out.println()** for console output.