

Java Learning Journey

Chapter 4: Mathematical Functions, Characters, and Strings

 **Prepared by:** Ahmed Elsifi

1. Math Class Methods and Constants

The `Math` class provides common mathematical functions and constants. All methods are `static`, so they are called using `Math.methodName()`.

Constants:

- `Math.PI` – π (approx. 3.141592653589793)
- `Math.E` – base of natural logarithms (approx. 2.718281828459045)

Trigonometric Methods:

- `Math.sin(radians)` – returns the sine of an angle in radians
- `Math.cos(radians)` – returns the cosine of an angle in radians
- `Math.tan(radians)` – returns the tangent of an angle in radians
- `Math.toRadians(degrees)` – converts degrees to radians
- `Math.toDegrees(radians)` – converts radians to degrees
- `Math.asin(a)` – returns arc sine in radians ($-\pi/2$ to $\pi/2$)
- `Math.acos(a)` – returns arc cosine in radians (0 to π)
- `Math.atan(a)` – returns arc tangent in radians ($-\pi/2$ to $\pi/2$)

Exponent Methods:

- `Math.exp(x)` – returns e raised to the power x (e^x)
- `Math.log(x)` – returns natural logarithm ($\ln x$)
- `Math.log10(x)` – returns base-10 logarithm ($\log_{10} x$)
- `Math.pow(a, b)` – returns a raised to the power b (a^b)
- `Math.sqrt(x)` – returns square root of x (\sqrt{x})

Rounding Methods:

- `Math.ceil(x)` – rounds up to nearest integer (as double)
- `Math.floor(x)` – rounds down to nearest integer (as double)
- `Math rint(x)` – rounds to nearest integer (if equidistant, returns even integer)
- `Math.round(x)` – returns closest int (for float) or long (for double)

Other Methods:

- `Math.max(a, b)` – returns the larger of two values
- `Math.min(a, b)` – returns the smaller of two values
- `Math.abs(x)` – returns absolute value
- `Math.random()` – returns random double in [0.0, 1.0)

Generating Random Numbers Between x and y:

```
int randomNum = x + (int)(Math.random() * (y - x + 1));
```

2. Character Class Methods

The `Character` class provides static methods for testing and converting characters:

- `Character.isDigit(ch)` – returns true if `ch` is a digit
 - `Character.isLetter(ch)` – returns true if `ch` is a letter
 - `Character.isLetterOrDigit(ch)` – returns true if `ch` is a letter or digit
 - `Character.isLowerCase(ch)` – returns true if `ch` is lowercase
 - `Character.isUpperCase(ch)` – returns true if `ch` is uppercase
 - `Character.toLowerCase(ch)` – returns lowercase version of `ch`
 - `Character.toUpperCase(ch)` – returns uppercase version of `ch`
-

3. String Data Type and Methods

Strings are objects in Java (reference type). Common methods:

Basic Methods:

- `length()` – returns number of characters
- `charAt(index)` – returns character at specified index
- `concat(s)` – concatenates strings (same as `+`)
- `toLowerCase()` – returns new lowercase string
- `toUpperCase()` – returns new uppercase string
- `trim()` – returns string with whitespace removed from both ends

Comparison Methods:

- `equals(s)` – returns true if contents are equal
- `equalsIgnoreCase(s)` – case-insensitive equality check
- `compareTo(s)` – returns 0 if equal, <0 if lexicographically less, >0 if greater
- `compareToIgnoreCase(s)` – case-insensitive comparison
- `startsWith(prefix)` – returns true if string starts with prefix
- `endsWith(suffix)` – returns true if string ends with suffix
- `contains(s)` – returns true if substring exists

Substring and Search Methods:

- `substring(beginIndex)` – returns substring from `beginIndex` to end
- `substring(beginIndex, endIndex)` – returns substring from `beginIndex` to `endIndex-1`
- `indexOf(ch)` – returns first occurrence index of character
- `indexOf(ch, fromIndex)` – returns occurrence index after `fromIndex`
- `indexOf(s)` – returns first occurrence index of substring
- `indexOf(s, fromIndex)` – returns substring occurrence after `fromIndex`
- `lastIndexOf(...)` – similar to `indexOf` but searches backward

Conversion:

- `Integer.parseInt(str)` – converts string to int
- `Double.parseDouble(str)` – converts string to double

4. Non-Instance vs. Static Methods

- **Instance methods:** Called on an object (e.g., `str.length()`). Operate on instance data.
- **Static methods:** Called on class (e.g., `Math.sqrt(4)`). Don't require object instance.

5. String Comparison Methods

- Use `equals()` for content equality (not `==`).
- Use `compareTo()` for lexicographical ordering.
- Use `equalsIgnoreCase()` and `compareToIgnoreCase()` for case-insensitive comparisons.

6. Format Specifiers

Used in `System.out.printf()` for formatted output:

Specifier	Output
<code>%b</code>	Boolean
<code>%c</code>	Character
<code>%d</code>	Decimal integer
<code>%f</code>	Floating-point
<code>%e</code>	Scientific notation
<code>%s</code>	String

Width and Alignment:

- `%10d` – right-align in 10-character width
- `%-10d` – left-align in 10-character width
- `%10.2f` – 10-width with 2 decimal places
- `%,d` – adds comma separators (e.g., 1,000)
- `%08d` – pads with zeros to 8 digits (e.g., 000123)

Examples:

```
System.out.printf("%-10s%10d", "Name", 25); // Left-align string, right-align number
```