7. String Handling

Lesson 2:
String manipulation methods

7.2. String Manipulation Methods

7.2.1. Constructors

- The String class has seven different constructors. We describe some of these.

- The simplest method to create a String object is to enclose the string literal in quotes and assign the value to a String object. Creation of a String through assignment does not require the use of the new operator, for example:

```
String str = "abc";
```

- Alternatively, String objects can be created through constructors. The following constructors are supported:

public String()

  - Constructs a new String with the value "" containing no characters. The value is not null.

public String(String value)

  - Constructs a new String that contains the same sequence of characters as the specified String argument.

public String(char[] value)

  - Constructs a new String containing the same sequence of characters contained in the character array argument.

7.2. String Manipulation Methods

7.2.2. String Concatenation

- Java Strings can be concatenated (joined) using the + and += operators to create new Strings.

```
String language = "Java";
String course = "Introduction to " + language;
course += ", CS 101";
```

- Every time an operation modifies a String object, a new read-only String object is created.

7.2. String Manipulation Methods

7.2.3. Comparing Strings

public int compareTo(String str)

- Compares the current String object to str, and returns 0 only if the two strings contain the same sequence of characters (case sensitive).
  - A negative value is returned if the current String is lower in the Unicode set than the String str.
  - A positive value is returned if the current String is higher in the Unicode set than the String str.

public boolean equals(Object obj)

- Compares the current String object with obj and returns true if obj is another String containing the same sequence of characters; false is returned otherwise.

public boolean equalsIgnoreCase(String str)

- Performs a case-insensitive comparison of the current String object with str and returns true if str is another String containing the same sequence (ignoring case) of characters; false is returned otherwise.

- Use of the **==** operator only tests whether two String object references refer to the same object (memory space).

7.2. String Manipulation Methods

7.2.4. Other useful methods

public char charAt(int index)

- Returns the character at the specified index. The index may range from 0 to length() - 1.

public String concat(String str)

- Concatenates the specified String to the end of the current (this) String. If the length of the argument string is 0, then this String is returned.

public int length()

- Returns the length of the current String. The length is equal to the number of 16-bit Unicode characters in the String.

public String toUpperCase()

- Converts the String to uppercase.

public String toLowerCase()

- Converts the String to lowercase

7.2. String Manipulation Methods

7.2.5. More useful methods

public String trim()

- Removes white space from both ends of the String. All characters that have codes less than or equal to '\u0020' (the space character) are considered to be white space.

public static String valueOf(boolean b)
public static String valueOf(char ch)
public static String valueOf(int inum)
public static String valueOf(long lnum)
public static String valueOf(float fnum)
public static String valueOf(double dnum)

- Creates the String representation of the argument.