Java String

Introduction to Strings in Java

- **Definition**: A string in Java is an object that represents a sequence of characters.
- **Immutability**: Strings in Java are immutable, meaning once a string object is created, its content cannot be changed. This is handled internally by creating a new string object whenever a modification is made.

Creating Strings

- **Using String Literals**:
  ```
  String str1 = "Hello, World!";
  ```
- **Using the `new` Keyword**:
  ```
  String str2 = new String("Hello, World!");
  ```

String Methods

- **Length**:
  ```
  int length = str1.length();
  ```
- **Character Extraction**:
  ```
  char ch = str1.charAt(1); // 'e'
  ```
- **Substring**:
  ```
  String substr = str1.substring(0, 5); // "Hello"
  ```
- **String Comparison**:
  ```
  boolean isEqual = str1.equals(str2); // true if str1 and str2 are the same
  boolean isEqualIgnoreCase = str1.equalsIgnoreCase("HELLO, WORLD!"); // true
  int comparisonResult = str1.compareTo(str2); // 0 if equal, negative if str1 < str2, positive if str1 > str2
  ```
- **Searching in Strings**:
  ```
  int index = str1.indexOf('o'); // 4
  int lastIndex = str1.lastIndexOf('o'); // 8
  boolean contains = str1.contains("World"); // true
  ```

- **Replacing Characters/Substrings**:
  ```
  String replacedStr = str1.replace('o', 'a'); // "Hella, Warld!"
  String replacedStr2 = str1.replace("World", "Java"); // "Hello,
  Java!"
  ```

- **Case Conversion**:
  ```
  String upperCaseStr = str1.toUpperCase(); // "HELLO, WORLD!"
  String lowerCaseStr = str1.toLowerCase(); // "hello, world!"
  ```

- **Trimming Whitespace**:
  ```
  String trimmedStr = "  Hello, World!  ".trim(); // "Hello,
  World!"
  ```

String Formatting

- **Using `String.format`**:
  ```
  String formattedStr = String.format("Hello, %s! You have %d new
  messages.", "Alice", 5);
  // "Hello, Alice! You have 5 new messages."
  ```

Regular Expressions with Strings

- **Pattern Matching**:
  ```
  String regex = "\\d+";
  boolean matches = str1.matches(regex); // true if str1 contains
  only digits
  ```
- **Splitting Strings**:
  ```
  String[] words = str1.split(", ");
  // words = ["Hello", "World!"]
  ```

String Pool

- Java maintains a pool of strings for efficient memory usage. String literals are interned, meaning they are stored in a common pool.
  ```
  String str3 = "Hello";
  String str4 = "Hello";
  boolean isSameReference = (str3 == str4); // true, because both
  refer to the same object in the string pool
  ```

Common String Manipulation Examples

- **Reversing a String**:
  ```
  public String reverseString(String input) {
       return new StringBuilder(input).reverse().toString();
  ```

```
}
```

- **Checking for Palindromes**:

```
public boolean isPalindrome(String input) {
    int left = 0;
    int right = input.length() - 1;
    while (left < right) {
        if (input.charAt(left) != input.charAt(right)) {
            return false;
        }
        left++;
        right--;
    }
    return true;
}
```