Our Solution(s)   Run Code

Solution 1   Solution 2

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // O(n^2) time | O(1) space
  public static String longestPalindromicSubstring(String str)
    int[] currentLongest = {0, 1};
    for (int i = 1; i < str.length(); i++) {
      int[] odd = getLongestPalindromeFrom(str, i - 1, i + 1);
      int[] even = getLongestPalindromeFrom(str, i - 1, i);
      int[] longest = odd[1] - odd[0] > even[1] - even[0] ? odd
      currentLongest =
          currentLongest[1] - currentLongest[0] > longest[1] -
              ? currentLongest
              : longest;
    }
    return str.substring(currentLongest[0], currentLongest[1]);
  }

  public static int[] getLongestPalindromeFrom(String str, int
    while (leftIdx >= 0 && rightIdx < str.length()) {
      if (str.charAt(leftIdx) != str.charAt(rightIdx)) {
        break;
      }
      leftIdx--;
      rightIdx++;
    }
    return new int[] {leftIdx + 1, rightIdx};
  }
}
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```java
class Program {
  public static String longestPalindromicSubstring(String str) {
    // Write your code here.
    return "";
  }
}
```

Our Tests   Custom Output   Submit Code

Run or submit code when you're ready.