JAVA.LANG.CHARACTER.OFFSETBYCODEPOINTS METHOD

Description

The **java.lang.Character.offsetByCodePoints***char[]a, intstart, intcount, intindex, intcodePointOffset* returns the index within the given char subarray that is offset from the given index by codePointOffset code points.

The start and count arguments specify a subarray of the char array. Unpaired surrogates within the text range given by index and codePointOffset count as one code point each.

Declaration

Following is the declaration for **java.lang.Character.offsetByCodePoints** method

```
public static int offsetByCodePoints(char[] a, int start, int count, int index, int
codePointOffset)
```

Parameters

- **a** - the char array
- **start** - the index of the first char of the subarray
- **count** - the length of the subarray in chars
- **index** - the index to be offset
- **codePointOffset** - the offset in code points

Return Value

This method returns the index within the subarray

Exception

- **NullPointerException** - if a is null
- **IndexOutOfBoundsException** - if start or count is negative, or if start + count is larger than the length of the given array, or if index is less than start or larger then start + count, or if codePointOffset is positive and the text range starting with index and ending with start + count - 1 has fewer than codePointOffset code points, or if codePointOffset is negative and the text range starting with start and ending with index - 1 has fewer than the absolute value of codePointOffset code points.

Example

The following example shows the usage of lang.Character.offsetByCodePoints method.

```
package com.tutorialspoint;

import java.lang.*;

public class CharacterDemo {

   public static void main(String[] args) {

      // create a char array c and assign values
      char[] c = new char[] { 'a', 'b', 'c', 'd', 'e', 'f' };

      // craete 2 int primitives start, count and assign values
      int start = 1;
      int count = 5;
```

```
      // create an int primitive res
      int res;

      // assign result of offsetByCodePoints on subarray of c to res
      res = Character.offsetByCodePoints(c, start, count, 2, 4);

      String str = "The index within the subarray of c is " + res;

      // print res value
      System.out.println( str );
   }
}
```

Let us compile and run the above program, this will produce the following result:

```
The index within the subarray of c is 6
```