JAVA.UTIL.BITSET.GET METHOD

Description

The **java.util.BitSet.get*intfromIndex, inttoIndex*** method returns a new BitSet composed of bits from this BitSet from fromIndex *inclusive* to toIndex *exclusive*.

Declaration

Following is the declaration for **java.util.BitSet.get** method

```
public BitSet get(int fromIndex,int toIndex)
```

Parameters

- **fromIndex** -- index of the first bit to include.
- **toIndex** -- index after the last bit to include.

Return Value

This method returns a new BitSet from a range of this BitSet.

Exception

- **IndexOutOfBoundsException** -- if fromIndex is negative, or toIndex is negative, or fromIndex is larger than toIndex.

Example

The following example shows the usage of java.util.BitSet.get method.

```
package com.tutorialspoint;

import java.util.*;

public class BitSetDemo {

   public static void main(String[] args) {

   // create 2 bitsets
   BitSet bitset1 = new BitSet(8);
   BitSet bitset2 = new BitSet(8);

   // assign values to bitset1
   bitset1.set(0);
   bitset1.set(1);
   bitset1.set(2);
   bitset1.set(3);
   bitset1.set(4);
   bitset1.set(5);

   // assign values to bitset2
   bitset2.set(2);
   bitset2.set(4);
   bitset2.set(6);
   bitset2.set(8);
   bitset2.set(10);

   // print the sets
   System.out.println("Bitset1:" + bitset1);
   System.out.println("Bitset2:" + bitset2);
```

```
   // get index 1 to 4 of bitset1
   System.out.println("" + bitset1.get(1,4));

   // get index 2 to 10 of bitset2
   System.out.println("" + bitset2.get(2,10));

   }
}
```

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

```
Bitset1:{0, 1, 2, 3, 4, 5}
Bitset2:{2, 4, 6, 8, 10}
{0, 1, 2}
{0, 2, 4, 6}
```