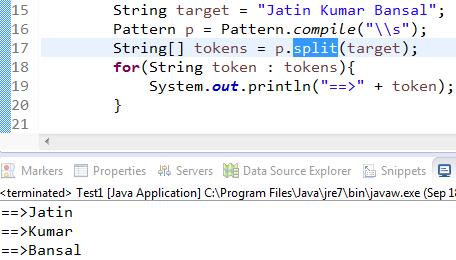
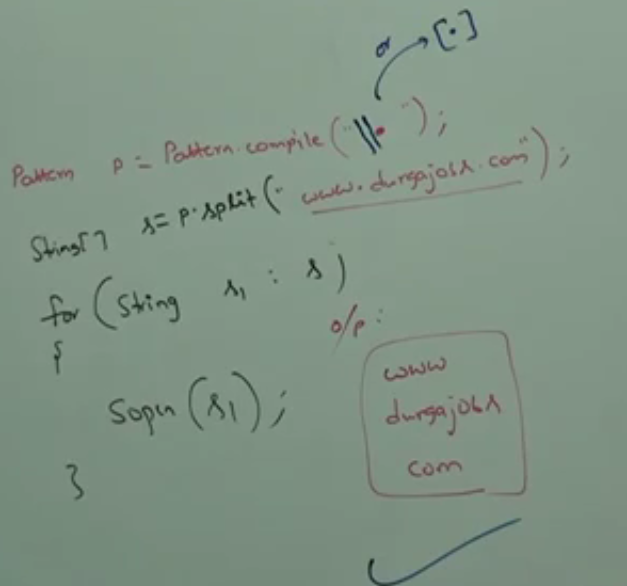
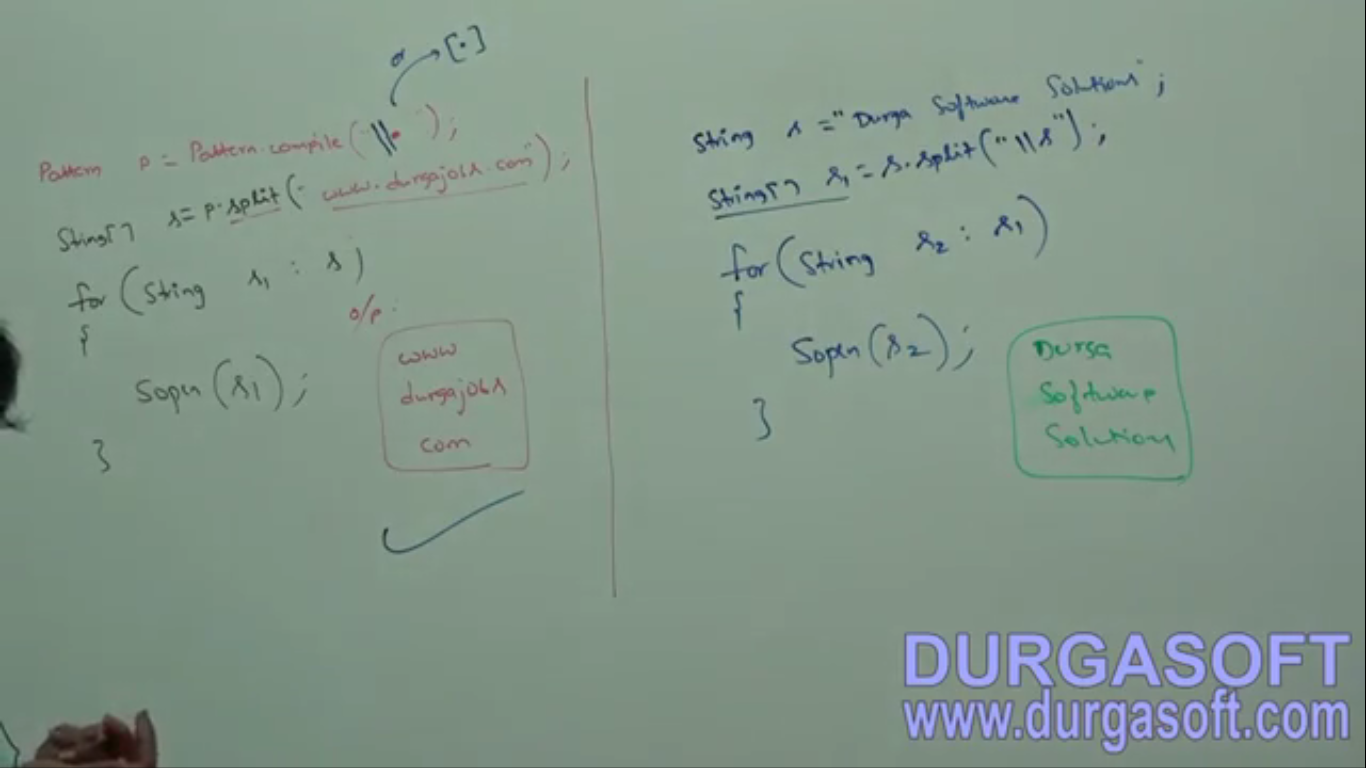
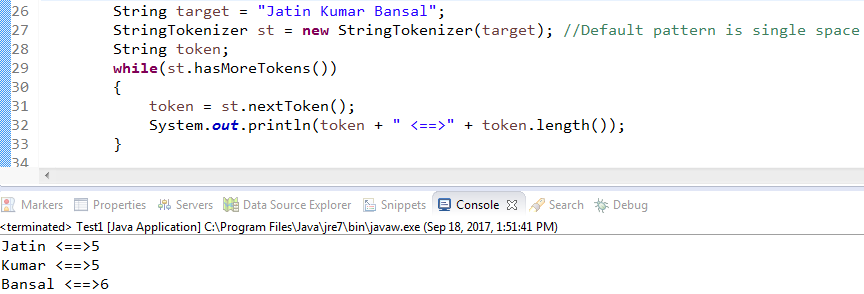
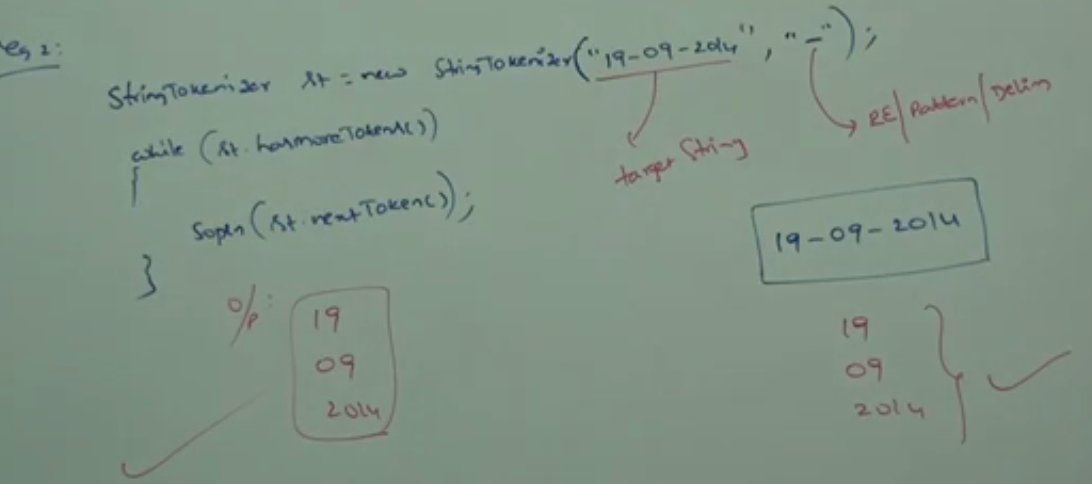
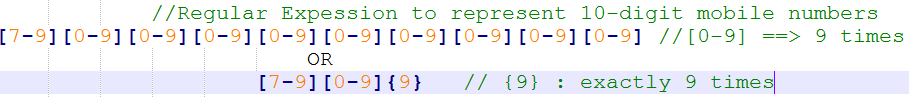
Pattern class split()

1. Used to split the target string according to a particular pattern.
2. **How does split() work?**
   1. Take a target String 🡺 “Jatin Kumar”; //**NOTE**: There are two spaces b/w “Jatin” and “Kumar”.
   2. Pattern 🡺 “[ ]” 🡺 Based on space. There is space in b/w.
   3. Now call split()
      1. it will search for patter which is a single space and where it finds space, it draws a slice. b/w the slice collects all characters and makes a token.
      2. NOTE: if in target string two or more continuous spaces are there then two or more slices will be continuous and then b/w the slices, there would be no character, in that case, it would create an empty string.
3. **Example**:  
     
     
   **NOTE:String class also contains split()** to split the target string according to a particular pattern.   
   **Example**:  
     
   **NOTE**: Pattern class **split(String target)** can take target string as argument where as Spring class split(**String pattern**) can take pattern as argument.
4. d

StringTokenizer

1. It is a specially designed class for tokenization activities.
2. StringTokenizer is present in 🡺 java.util
3. **Example**:  
     
   **NOTE**: The default regular expression for StringTokenizer is space.   
   **Example**: Setting your own regular expression (Delimiter or Pattern) explicitly.   
   

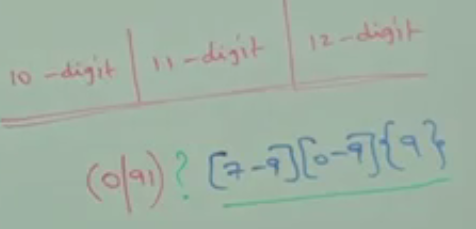
## Write a regular expression to represent all valid 10-digit mobile numbers.

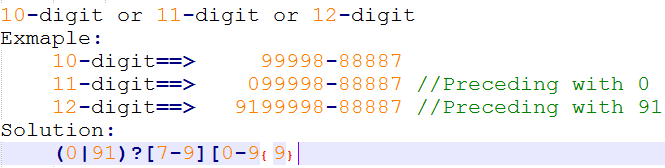


## Write a regular expression to represent either 10-digit or 11-digit mobile number where the 1st digit is optional which is zero (0) Example: 9998887776 or 09998887776 🡺 Both are same.

// ? means the preceding character zero or one time.

**Example**:





## Regular Expression to validate email format Write a regular expression to represent all java language identifiers with the following customized rules

First dot then alpha char any times and the sequence either 1 or more times

Alphanumeric character either 1 or more times

Then @

The next character must be alphanumeric with any number of times

The first character must be alphanumeric

