

JDLC2 FCS COMPUTATION EXERCISE

Version 1.005

PURPOSE

This purpose of this exercise is for you to demonstrate your understanding of cyclic redundancy check (CRC) by writing a method to compute the CRC frame check sequence (FCS) for a frame represented as a bit set.

EXERCISE PROCESS

Begin by installing JNWS as described in the User's Guide. Confirm it will compile and run.

You are provided with a template for your program in class Exercise: class JDLC2. The template also contains the algorithm for CRC-FCS computation, from *Understanding Internet Protocols* page 44. The class has three methods:

```
/**Reads a single line from standard input*/  
private static String ReadSingleLineFromStandardIO ( )  
  
/**Computes the CRC FCS for the bit set dataBitSet*/  
private static Utility.JNWBitSet computeFCS ( Utility.JNWBitSet dataBitSet )  
  
/**Runs the FCS computation.*/  
public static void main ( String [ ] args )
```

Method `JNWBitSet computeFCS ()` is only a template. Your assignment is to complete the methods in class JDLC2 which are needed to compute the FCS of the email String. You might want to print intermediate results of the FCS computation for debugging.

The `main()` method accepts an email String, displays the String, converts it to a JNWBitSet, displays the result, computes the FCS for the bit set, appends the FCS to the bit set, and displays the binary result.

Suggestions:

1. Since even simple Java IO can become complicated, use method `ReadSingleLineFromStandardIO ()` to read a String representing the email from the standard input.

2. Methods that you may need are:
 - `setJNWLength (int length)` set the length of the JNWBitSet
 - `getJNWLength ()` gets the length of the JNWBitSet
 - `setBitValue (int bitindex , int value)` set the bit at index bitIndex to 0 or 1
 - `int getBitValue (int bitindex)` gets the bit at index bitIndex as a 0 or 1
 - `shift (int numberOfBitsToShift)` shifts the BitSet left (positive) or right (negative) by numberOfBitsToShift bits
 - `setJNWBitSetToString (String characterString)` converts a character String into a JNWBitSet
 - `toString ()` converts a JNWBitSet to a String of 0s and 1s 40 bits per row
 - `boolean java.util.BitSet.get (int bitIndex)` returns the bitIndexth bit in the bit set as a boolean (true or false) rather than a 0 or 1 as with `getBitValue (int bitindex)`
 - `java.util.BitSet.flip (int bitIndex)` sets the bit at the specified index to the complement of its current value.

3. Another useful method from class `java.util.BitSet` is

void	4. <code>xor (BitSet set)</code> Performs a logical XOR of this bit set with the bit set argument.
------	--

5. *This* bit set and the argument bit set *set* need not be the same length. XOR starts with the rightmost (bitIndex = 0) bit and performs the xor function until it reaches the end of either *this* or *set*. However, if you use this method be aware that you need to XOR with the FCS the feedback bit, not the CRC polynomial.
6. To truncate a JNWBitSet at the low order end use `shift(-numberOfBitsToTruncate)`. To truncate a JNWBitSet at the high order end use `setJNWLength (JNWLength – numberOfBitsToTruncate)`.
7. During testing some students have found it to be very useful to display the feedback bit, the FCS bits and the next input. You might want to combine this with running the "123" input from page 47 of *Understanding Internet Protocols*.
8. If you use NetBeans, you should review the advice in the User's Guide about setting the main () method to be run.
9. You can avoid confusion by using the convention of figure 4.4 from *Understanding Internet Protocols*, where the low order but (bit number 0) of the JNWBitString is envisioned to be on its right end. This works well with `JNWBitString.toString()`, which prints the bit string in that order.

EXERCISE

Write `computeFCS (Utility.JNWBitSet dataBitSet)`. Run the `main()` program for the email String "Chicken Little was right~" which is the text that will be used by the grader. Use the CCITT CRC polynomial, "1 00010000 00100001 "

DEMONSTRATION

The .jar file for Demonstrations.JDLC2_FCSComputation is available for you to see a demonstration. Download the file FCSComputation.jar and type, at the command prompt:

```
java -jar JDLC2_FCSComputation.jar
```

to run the demonstration.

SUBMITTING

Cut and paste the output from the program to a file. Send your output and file `Exercises.JDLC2.java` to the grader as email attachments.