Example.java

```java
import java.io.ByteArrayInputStream;

public class Example {

    /*
     * Input sequences for alignment
     */
    static final String input = ">Foo\r\n"
            + "MTADGPRELLQLRAAVRHRPQDFVAWLMLADAELGMGDTTAGEMAVQRGLALHPGHPEAVARLGR"
            + "VRWTQQRHAEAAVLLQQASDAAPEHPGIALWLGHALEDAGQAEAAAAYTRAHQLLPEEPYITAQ"
            + "LLNWRRRLCDWRALDVLSAQVRAAVAQGVGAVEPFAFLSEDASAAEQLACARTRAQAIAASVRPL"
            + "APTRVRSKGPLRVGFVSNGFGAHPTGLLTVALFEALQRRQPDLQMHLFATSGDDGSTLRTRLAQA"
            + "STLHDVTALGHLATAKHIRHHGIDLLFDLRGWGGGGRPEVFALRPAPVQVNWLAYPGTSGAPWMD"
            + "YVLGDAFALPPALEPFYSEHVLRLQGAFQPSDTSRVVAEPPSRTQCGLPEQGVVLCCFNNSYKLN"
            + "PQSMARMLAVLREVPDSVLWLLSGPGEADARLRAFAHAQGVDAQRLVFMPKLPHPQYLARYRHAD"
            + "LFLDTHPYNAHTTASDALWTGCPVLTTPGETFAARVAGSLNHHLGLDEMNVADDAAFVAKAVALAS"
            + "DPAALTALHARVDVLRRESGVFEMDGFADDFGALLQALARRHGWLGI\r\n"
            + "\r\n"
            + ">Bar\r\n"
            + "MGDTTAGEMAVQRGLALHQQRHAEAAVLLQQASDAAPEHPGIALWLHALEDAGQAEAAAAYTRAH"
            + "QLLPEEPYITAQLLNAVAQGVGAVEPFAFLSEDASAAESVRPLAPTRVRSKGPLRVGFVSNGFGA"
            + "HPTGLLTVALFEALQRRQPDLQMHLFATSGDDGSTLRTRLAQASTLHDVTALGHLATAKHIRHHG"
            + "IDLLFDLRGWGGGGRPEVFALRPAPVQVNWLAYPGTSGAPWMDYVLGDAFALPPALEPFYSEHVL"
            + "RLQGAFQPSDTSRVVAEPPSRTQCGLPEQGVVLCCFNNSYKLNPQSMARMLAVLREVPDSVLWLL"
            + "SGPGEADARLRAFAHAQGVDAQRLVFMPKLPHPQYLARYRHADLFLDTHPYNAHTTASDALWTGC"
            + "PVLTTPGETFAARVAGSLNHHLGLDEMNVADDAAFVAKAVALASDPAALTALHARVDVLRRESGV"
            + "FEMDGFADDFGALLQALARRHGWLGI\r\n"
            + "\r\n"
            + ">Friends\r\n"
            + "MTADGPRELLQLRAAVRHRPQDVAWLMLADAELGMGDTTAGEMAVQRGLALHPGHPEAVARLGRV"
            + "RWTQQRHAEAAVLLQQASDAAPEHPGIALWLGHALEDHQLLPEEPYITAQLDVLSAQVRAAVAQG"
            + "VGAVEPFAFLSEDASAAEQLACARTRAQAIAASVRPLAPTRVRSKGPLRVGFVSNGFGAHPTGLL"
            + "TVALFEALQRRQPDLQMHLFATSGDDGSTLRTRLAQASTLHDVTALGHLATAKHIRHHGIDLLFD"
            + "LRGWGGGGRPEVFALRPAPVQVNWLAYPGTSGAPWMDYVLGDAFALPPALEPFYSEHVLRLQGAF"
            + "QPSDTSRVVAEPPSRTQCGLPEQGVVLCCFNNSYKLNPQSMARMLAVLREVPDSVLWLLSGPGEA"
            + "DARLRAFAHAQGVDAQRLVFMPKLPHPQYLARYRHADLFLDTHPYNAHTTASDALWTGCPVLTTP"
            + "GETFAARVAGSLNHHLGLDEMNVADDAAFVAKAVALASDPAALTALHARVDVLRRESI";

    public static void main(String[] args) throws UnsupportedRuntimeException,
            LimitExceededException, JobSubmissionException,
            WrongParameterException, FileNotFoundException, IOException,
            ResultNotAvailableException, InterruptedException {

        String qualifiedServiceName = "http://msa.data.compbio/01/01/2010/";

        /* Make a URL pointing to web service WSDL */
        URL url = new URL(
                "http://www.compbio.dundee.ac.uk/jabaws/ClustalWS?wsdl");

        /*
         * If you are making a client that connects to different web services
         * you can use something like this:
         */
        // URL url = new URL(host + "/" + Services.ClustalWS.toString() +
        // "?wsdl");

        QName qname = new QName(qualifiedServiceName, "ClustalWS");
        Service serv = Service.create(url, qname);
        /*
         * Multiple sequence alignment interface for Clustal web service
         * instance
         */
```

Example.java

```java
        MsaWS msaws = serv.getPort(new QName(qualifiedServiceName, "ClustalWS"
                + "Port"), MsaWS.class);

        /* Get the list of available presets */
        PresetManager presetman = msaws.getPresets();

        /* Get the Preset object by preset name */
        Preset preset = presetman
                .getPresetByName("Disable gap weighting (Speed-oriented)");

        /*
         * Load sequences in FASTA format from the file You can use something
         * like new FileInputStream(<filename>) to load sequence from the file
         */
        List<FastaSequence> fastalist = SequenceUtil
                .readFasta(new ByteArrayInputStream(input.getBytes()));

        /*
         * Submit loaded sequences for an alignment using preset The job
         * identifier is returned by this method, you can retrieve the results
         * with it sometime later.
         */
        String jobId = msaws.presetAlign(fastalist, preset);

        /* This method will block for the duration of the calculation */
        Alignment alignment = msaws.getResult(jobId);

        /*
         * This is a better way of obtaining results, it does not involve
         * holding the connection open for the duration of the calculation,
         * Besides, as the University of Dundee public server will reset the
         * connection after 10 minutes of idling, this is the only way to obtain
         * the results of long running task from our public server.
         */
        // while (msaws.getJobStatus(jobId) != JobStatus.FINISHED) {
        // Thread.sleep(1000); // wait a second, then recheck the status
        // }

        /* Output the alignment to standard out */
        System.out.println(alignment);

        // Alternatively, you can record retrieved alignment into the file in
        // ClustalW format

        // ClustalAlignmentUtil.writeClustalAlignment(new FileOutputStream(
        // "output.al"), alignment);

    }
}
```