SentenceTools-1.0	
Requirement	**Response**
Using appropriate coding language	I used Java because it is a versatile, widely used, object-oriented and extensible language. My code is compiled with a compliance level of Java-1.8 which ensures it is backwardly compatible with older versions of Java. I used: • Oracle JDK • Eclipse • Apache Maven • JUnit • Jaxws • Wildfly • Git • SoapUI
Ready to plug into software that ships to customers	I developed the project so it can be compiled into a jar file which can be included in other Java projects. I also built a SOAP web service which allows anyone to validate sentences from any programming language if they have an internet connection. The SOAP service is deployed here: http://getafix.garland.co.za:8787/validation/SentenceService?wsdl
Good structure	Project structure: • src/main: contains Java source code for project • src/test: contains Java source code for test cases • apidocs: contains documentation on the api calls • soap-tests: tests run from SoapUI I used maven to manage dependencies, running test cases and building the jar/ war (web archive).
Style	Standard Java code formatting and naming conventions https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html
| Proper use of language idioms | Eg<br>```java
for (int i = 0; i < sentence.length(); i++) {
```<br>instead of<br>```java
for (int i = 0; sentence.length() > i ; ++i) {
``` |
Testability	Code easily tested with Junit test cases. I used SoapUI to test the SOAP services, soap-tests/SOAP-UI-tests.xml https://www.soapui.org/downloads/soapui/
Unit tests	Implemented the 11 test cases provided, I also included an extra 6
Comments	API documentation is generated with Javadoc tags. The code is commented where I felt it needed to be.
Logging	java.util.logging.Logger statements to help with debugging. Not using log4j