# Lab Work

Configure your workstation so that we can use maven. You can use the “apache-maven-3.5.0” installation provided. You will need to set up your environment variables.

Create a new settings.xml file and use it to point to the maven central repository.

Configure your workspace to use the new setting.xml file.

* In your settings.xml:
  + Add a mirror to the link <https://repo.maven.apache.org/maven2/>
  + Configure the path of your local repository.

Create a simple maven project on eclipse IDE with:

Groupid : com.lab.maven

ArtifactId: mavproj

Create a class called **com.lab.maven.mavproj.FizzBuzz.java**

In this class, create a method “***fizzBuzzIt***” which takes an ***int*** as input.

The method needs to return:

* “Fizz” if the input value is divisible by 3
* “Buzz” if the input value is divisible by 5
* “FizzBuzz” if the input value is divisible by both 3 and 5

Create a test class called **com.test.lab.maven.mavenproj.FizzBuzzTest.java**

This class should test the “fizzBuzzIt” method created previously, like below:

|  |
| --- |
| package fizzbuzz;  import org.assertj.core.api.Assertions;  import org.junit.Test;  public class FizzbuzzTest extends TestCase {    public void should\_return\_1\_when\_fizzBuzz\_is\_called\_with\_1() {  // Given  int input = 1;  String expected = "1";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }    public void should\_return\_2\_when\_fizzBuzz\_is\_called\_with\_2() {  // Given  int input = 2;  String expected = "2";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }  public void should\_return\_Fizz\_when\_fizzBuzz\_is\_called\_with\_3() {  // Given  int input = 3;  String expected = "Fizz";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }  public void should\_return\_Fizz\_when\_fizzBuzz\_is\_called\_with\_6() {  // Given  int input = 6;  String expected = "Fizz";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }  public void should\_return\_Buzz\_when\_fizzBuzz\_is\_called\_with\_5() {  // Given  int input = 5;  String expected = "Buzz";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }  public void should\_return\_Buzz\_when\_fizzBuzz\_is\_called\_with\_10() {  // Given  int input = 10;  String expected = "Buzz";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }  public void should\_return\_FizzBuzz\_when\_fizzBuzz\_is\_called\_with\_15() {  // Given  int input = 15;  String expected = "FizzBuzz";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }    public void should\_return\_FizzBuzz\_when\_fizzBuzz\_is\_called\_with\_30() {  // Given  int input = 30;  String expected = "FizzBuzz";  FizzBuzz fizzBuzz = new FizzBuzz();  // When  String result = fizzBuzz.fizzBuzzIt(input);  // Then  Assertions.assertThat(result).isEqualTo(expected);  }  } |

1. Why are we having errors several lines?

Configure your pom.xml so that you can download into your local repository the jars “junit-4.12.jar” and “assert-core-3.7.0.jar”.

### Junit:

* groupId: junit
* artifactId: junit
* version: 4.12

### AssertJ Core:

* groupId: org.assertj
* artifactId: assert-core
* version: 3.7.0

1. Open your command line in your workspace and launch the specific maven commands which will enable us to download our jar files.
2. Launch your application again.