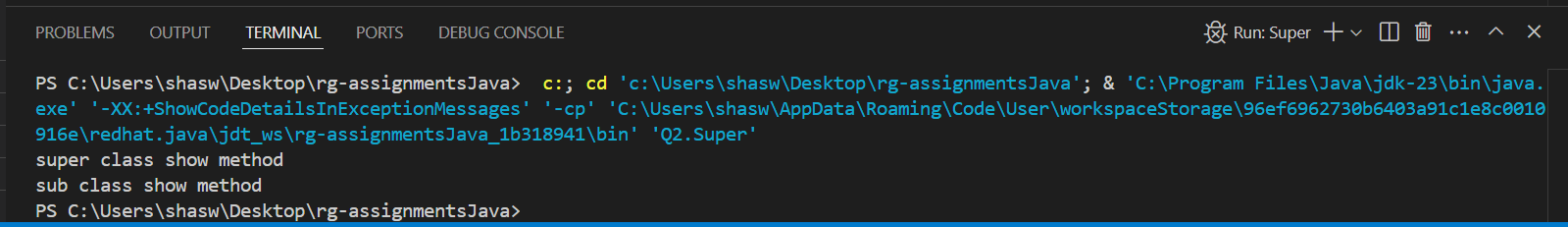
Q1 Explanation.

A pure function is a function which gives the same output for the same input and has no side effects. The method ‘TaxUtil’ is not pure because it depends on ‘rate’, which can be changed by other methods.

So, to make it pure, we can give all parameters which are affecting the output as the method parameters.

<https://github.com/sumshaswat/rg-assignments/tree/feature-java/rg-assignmentsJava/Q1>

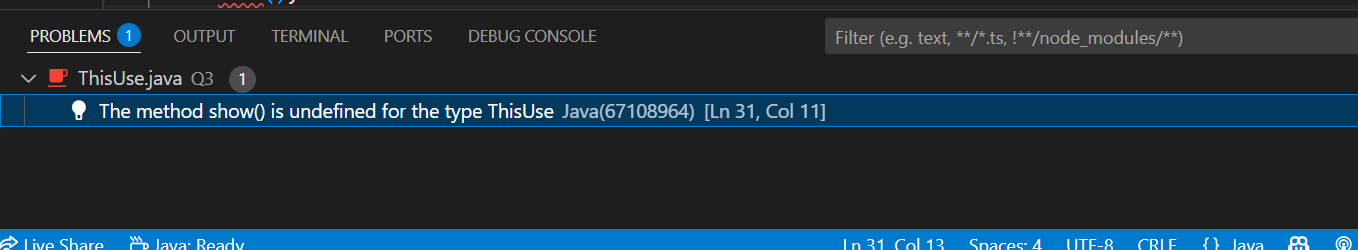
Q2 Explanation.



<https://github.com/sumshaswat/rg-assignments/tree/feature-java/rg-assignmentsJava/Q2>

Q3 Explanation.

show() method is not defined, instead Show() is defined, and Java is case sensitive, hence this code will give compilation errors as below:



<https://github.com/sumshaswat/rg-assignments/tree/feature-java/rg-assignmentsJava/Q3>

Q4 Explanation:

The Singleton design pattern ensures that only one instance of a class is created throughout the application. For Example:



1. private static DBConnection canObject = new DBConnection();

* private , so that cannot be used from outside, static so that only one instance can be created; no matter what.

private DBConnection(){}

* constructor made private, so that no other object from outside can be created

public static DBConnection getInstance() {

return canObject;

}

* If someone needs the object of this class, they can get through this function using class name(as the method id static).

<https://github.com/sumshaswat/rg-assignments/tree/feature-java/rg-assignmentsJava/Q4>

Q5 Explanation:

We can make sure the class is encapsulated by :

1. Making class variables private so they can't be accessed directly from outside.
2. Providing public to access and update the values safely.

<https://github.com/sumshaswat/rg-assignments/tree/feature-java/rg-assignmentsJava/Q5>

Q6 Explanation:

<https://github.com/sumshaswat/rg-assignments/tree/feature-java/rg-assignmentsJava/Q6>

Q7 Explanation:

<https://github.com/sumshaswat/rg-assignments/tree/feature-java/rg-assignmentsJava/Q7>