Data table is not same as example. Data table is not for data driven testing. It is for filling in data.

Scenario outline for data driven testing.

When we use scenario-outline we need to use examples keyword.

Examples keyword won't come with normal scenario.

"Examples" keyword used to achieve data driven.

Sometimes if the feature file is not showing the glued flags (orange highlight), right click on project and convert to cucumber or update the maven project, which we have done already many times.

Shortcut to import everything in a page is "control+shift+o".

Move the mouse over the (given, when) statements, which we have written etc and press "control + click" and you go inside the step def file.

Code with basic examples and scenario outline-

Feature file-

```
Feature: login feature

Scenario Outline: failed login - different combinations
	Given user is on application landing page
	When user clicks on signin button
	Then user is displayed login page
	When user enters "<userName>" in username field
	When user enters "<password>" in password field
	And user clicks on signin button
	Then error displayed for wrong credentials

		Examples:
			| userName | password |
			| incorrectusername | 234324 |
			| naveen auto | incorrectpassword |
			| incorrectusername | incorrectpassword |
```

Step def-

```
package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class LoginFeatureStepDef {

	@Given("user is on application landing page")
	public void user_is_on_application_landing_page() {

	}

	@When("user clicks on signin button")
	public void user_clicks_on_signin_button() {

	}

	@Then("user is displayed login page")
	public void user_is_displayed_login_page() {
```

```
	}

	@When("user enters {string} in username field")
	public void user_enters_in_username_field(String string) {

	}

	@When("user enters {string} in password field")
	public void user_enters_in_password_field(String string) {

	}

	@Then("error displayed for wrong credentials")
	public void error_displayed_for_wrong_credentials() {

	}

}
```

Output-

```
Scenario Outline: failed login - different combinations
  Given user is on application landing page
  When user clicks on signin button
  Then user is displayed login page
  When user enters "incorrectusername" in username field
(java.lang.String)
  When user enters "234324" in password field
(java.lang.String)
  And user clicks on signin button
  Then error displayed for wrong credentials

Scenario Outline: failed login - different combinations
  Given user is on application landing page
  When user clicks on signin button
  Then user is displayed login page
  When user enters "naveen auto" in username field
(java.lang.String)
  When user enters "incorrectpassword" in password field
(java.lang.String)
  And user clicks on signin button
  Then error displayed for wrong credentials
```

```
Scenario Outline: failed login - different combinations  #
  Given user is on application landing page               #
  When user clicks on signin button                       #
  Then user is displayed login page                       #
  When user enters "incorrectusername" in username field #
(java.lang.String)
  When user enters "incorrectpassword" in password field #
(java.lang.String)
  And user clicks on signin button                        #
  Then error displayed for wrong credentials              #

3 Scenarios (3 passed)
21 Steps (21 passed)
0m0.287s
```

Now see this –

We have same numeric field taking int and decimals.

```
2
3  Scenario Outline: bill amount
4    Given user is on billing page
5    When user enters bill amount <billAmount>
6    When user enters tax amount <taxAmount>
7    And user clicks calculate button
8    Then final amount is given <finalAmount>
9
10     Examples:
11       | billAmount | taxAmount | finalAmount |
12       | 1000   | 10 | 1010 |
13       | 100 | 40 | 140 |
14       | 20 | 6.7 | 26.7 |
```

In step def there will be overloaded method with int as param and double as param.

```
19      }
20
21      @When("user enters tax amount {int}")
22      public void user_enters_tax_amount(Integer int1) {
23          // Write code here that turns the phrase above into concrete actions
24          throw new io.cucumber.java.PendingException();
25      }
26
27      @When("user enters tax amount {double}")
28      public void user_enters_tax_amount(Double double1) {
29          // Write code here that turns the phrase above into concrete actions
30          throw new io.cucumber.java.PendingException();
31      }
32
```

In double we can store integer as well as double.

Let's see issue with cucumber which Naveen reported but cucumber guys and girls told it's the actual working-

Billing feature-

```
Feature: calculate billing amount

Scenario Outline: bill amount
      Given user is on billing page
      When user enters bill amount <billAmount>
      When user enters tax amount <taxAmount>
      And user clicks calculate button
      Then final amount is given <finalAmount>

            Examples:
                  | billAmount | taxAmount | finalAmount |
                  | 1000       | 10 | 1010 |
                  | 100 | 40 | 140 |
                  | 20 | 6.7 | 26.7 |
```

Billing step def-

```
package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import junit.framework.Assert;

public class BillingStepDef {

      int billingAmount;
      double taxAmount;
      double finalAmount;

      @Given("user is on billing page")
      public void user_is_on_billing_page() {

      }

      @When("user enters bill amount {int}")
      public void user_enters_bill_amount(Integer billingAmount) {
          this.billingAmount=billingAmount;
      }

      @When("user enters tax amount {int}")
      public void user_enters_tax_amount(Integer taxAmount) {
            this.taxAmount=taxAmount;
      }

      @When("user enters tax amount {double}")
      public void user_enters_tax_amount(Double taxAmount) {
            this.taxAmount=taxAmount;
      }

      @When("user clicks calculate button")
      public void user_clicks_calculate_button() {

      }

      @Then("final amount is given {int}")
      public void final_amount_is_given(Integer finalAmount) {
         this.finalAmount=this.billingAmount+this.taxAmount;
         Assert.assertTrue(this.finalAmount==finalAmount); // if the
condition is true then assertion should be passed
      }
```

```
	@Then("final amount is given {double}")
	public void final_amount_is_given(Double finalAmount) {
		this.finalAmount=this.billingAmount+this.taxAmount;
		Assert.assertTrue(this.finalAmount==finalAmount);
	}

}
```

Run feature and we get below exception-

It says there are multiple step def's which are matching. This is because integer can be stored in double also and it is confused which one to select.

With cucumber 4.0 it was working fine, from cucumber 6 this issue comes.

```
When user enters tax amount 40 # null
io.cucumber.core.runner.AmbiguousStepDefinitionsException: "user enters
tax amount 40" matches more than one step definition:
"user enters tax amount {double}" in
StepDefinitions.BillingStepDef.user_enters_tax_amount(java.lang.Double)
"user enters tax amount {int}" in
StepDefinitions.BillingStepDef.user_enters_tax_amount(java.lang.Integer)
at
io.cucumber.core.runner.CachingGlue.findStepDefinitionMatch(CachingGlue.j
ava:373)
at
io.cucumber.core.runner.CachingGlue.stepDefinitionMatch(CachingGlue.java:
341)
at
io.cucumber.core.runner.Runner.matchStepToStepDefinition(Runner.java:146)
at
io.cucumber.core.runner.Runner.createTestStepsForPickleSteps(Runner.java:
126)
at
io.cucumber.core.runner.Runner.createTestCaseForPickle(Runner.java:109)
at io.cucumber.core.runner.Runner.runPickle(Runner.java:70)
at io.cucumber.core.runtime.Runtime.lambda$execute$5(Runtime.java:110)
at
io.cucumber.core.runtime.CucumberExecutionContext.runTestCase(CucumberExe
cutionContext.java:117)
at io.cucumber.core.runtime.Runtime.lambda$execute$6(Runtime.java:110)
at
java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at
io.cucumber.core.runtime.Runtime$SameThreadExecutorService.execute(Runtim
e.java:233)
at
java.util.concurrent.AbstractExecutorService.submit(AbstractExecutorServi
ce.java:112)
at io.cucumber.core.runtime.Runtime.lambda$run$2(Runtime.java:86)
at
java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:193)
at java.util.stream.SliceOps$1$1.accept(SliceOps.java:204)
at
java.util.ArrayList$ArrayListSpliterator.tryAdvance(ArrayList.java:1359)
at
java.util.stream.ReferencePipeline.forEachWithCancel(ReferencePipeline.ja
va:126)
```

```
at
java.util.stream.AbstractPipeline.copyIntoWithCancel(AbstractPipeline.java:498)
at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:485)
at
java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)
at
java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:708)
at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
at java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:499)
at io.cucumber.core.runtime.Runtime.run(Runtime.java:87)
at io.cucumber.core.cli.Main.run(Main.java:92)
at cucumber.api.cli.Main.run(Main.java:30)
at cucumber.api.cli.Main.main(Main.java:15)
```

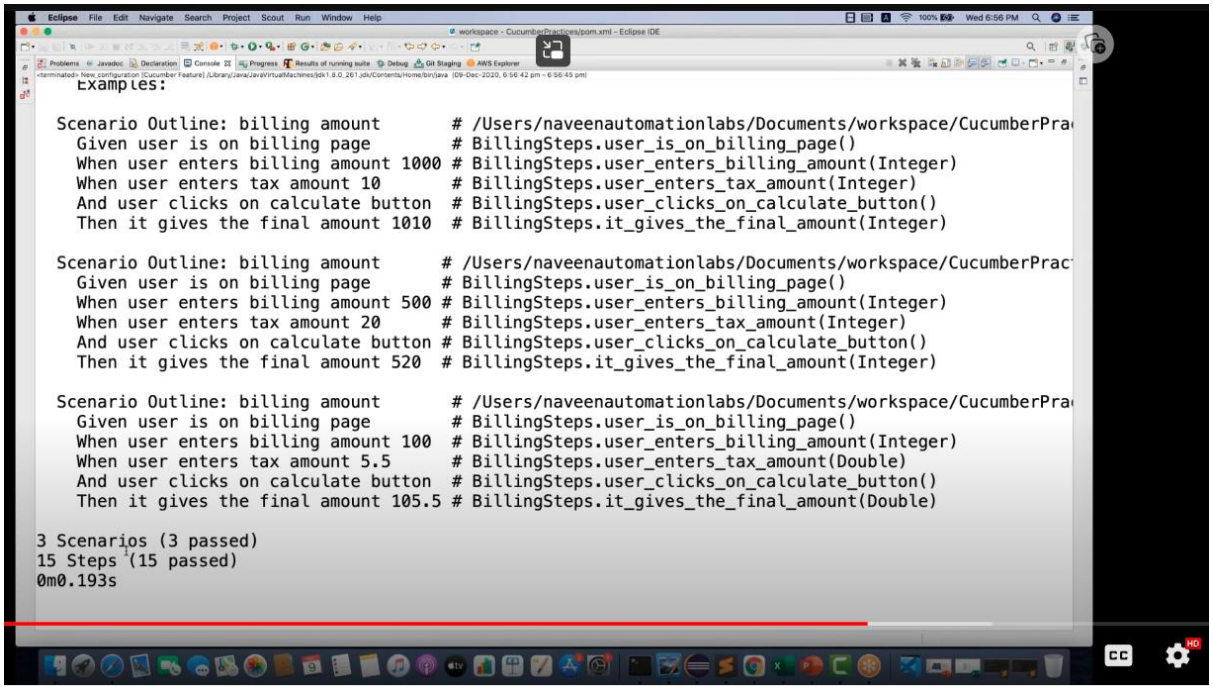
See with cucumber 4 –

Change in pom file the version of cucumber.

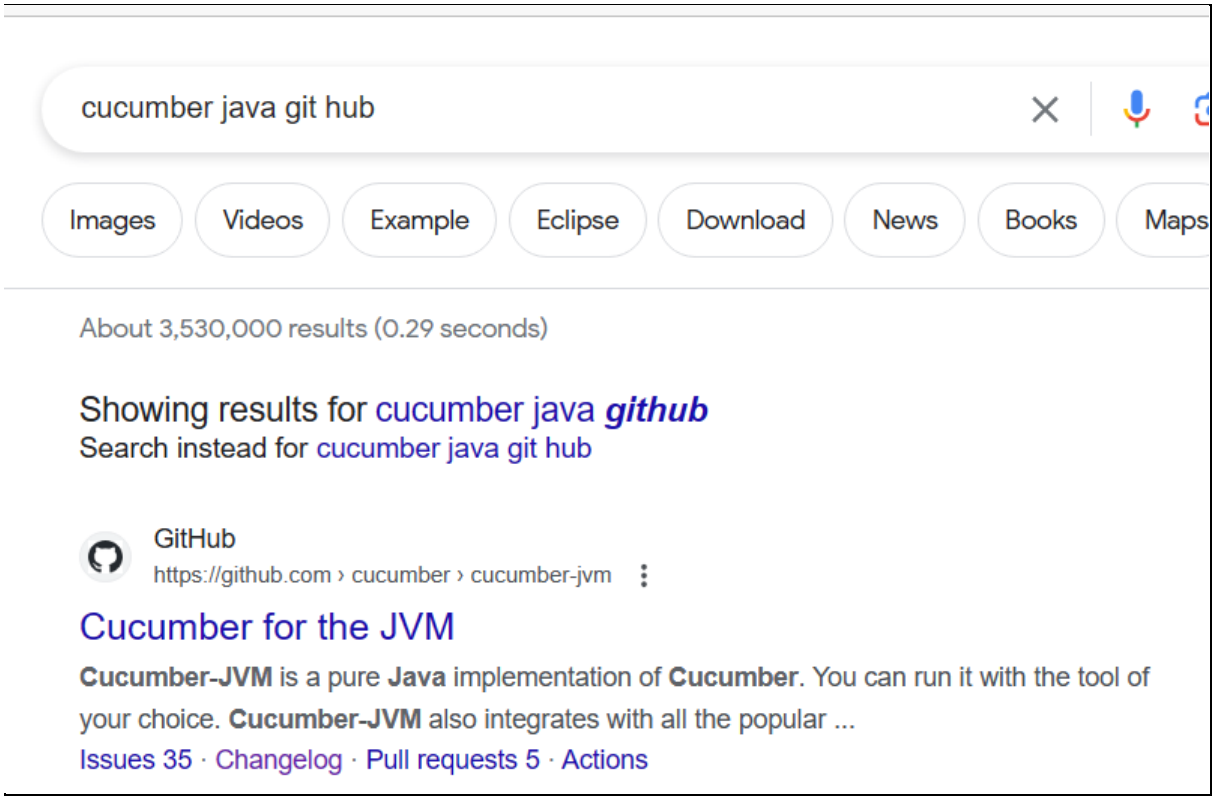
```
15
16      <properties>
17          <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
18          <java.version>1.8</java.version>
19          <junit.version>4.13.1</junit.version>
20          <cucumber.version>4.8.1</cucumber.version>
21          <maven.compiler.version>3.8.1</maven.compiler.version>
22          <maven.surefire.version>2.22.2</maven.surefire.version>
23      </properties>
24
```

Save it and it will rebuild.

Run feature and it works.



This is the official cucumber git hub page-



To solve the issue of overloaded methods use the input always in form of string in real time coding-

Billing feature-

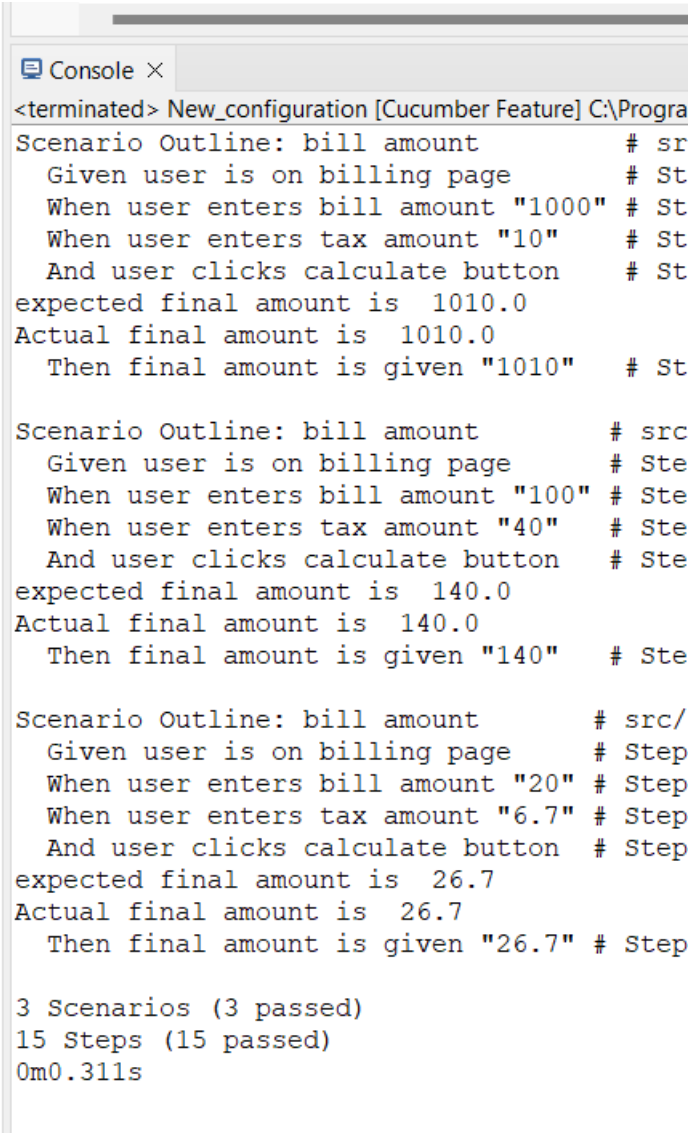
```
Feature: calculate billing amount

Scenario Outline: bill amount
      Given user is on billing page
      When user enters bill amount "<billAmount>"
      When user enters tax amount "<taxAmount>"
      And user clicks calculate button
      Then final amount is given "<finalAmount>"

            Examples:
                  | billAmount | taxAmount | finalAmount |
                  | 1000       | 10 | 1010 |
                  | 100 | 40 | 140 |
                  | 20 | 6.7 | 26.7 |
```

Step def-

```
package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import junit.framework.Assert;

public class BillingStepDef {

      double billingAmount;
```

```
	double taxAmount;
	double finalAmount;

	@Given("user is on billing page")
	public void user_is_on_billing_page() {

	}

	@When("user enters bill amount {string}")
	public void user_enters_bill_amount(String billingAmount) {
	    this.billingAmount=Double.parseDouble(billingAmount); //convert
string to double and give to the numeric guy/girl
	}

	@When("user enters tax amount {string}")
	public void user_enters_tax_amount(String taxAmount) {
		this.taxAmount=Double.parseDouble(taxAmount);
	}

	@When("user clicks calculate button")
	public void user_clicks_calculate_button() {

	}

	@Then("final amount is given {string}")
	public void final_amount_is_given(String finalAmount) {
	    this.finalAmount=this.billingAmount+this.taxAmount;
//this.finalAmount is already double as we have declared it.
	    System.out.println("expected final amount is " + " " +
Double.parseDouble(finalAmount));
	    System.out.println("Actual final amount is " + " " +
this.finalAmount);

Assert.assertTrue(this.finalAmount==Double.parseDouble(finalAmount)); //
if the condition is true then assertion should be passed
	}

}
```

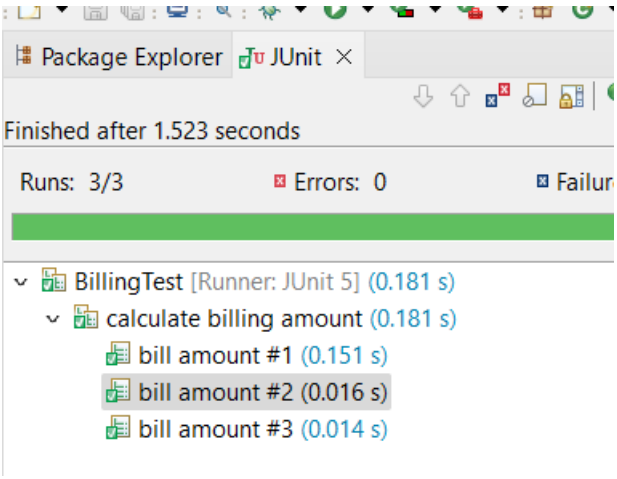
Output-



Runner file for billing-

```
package testRunners;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(
            plugin= {"pretty"},
            features = {"src/test/resources/AppFeatures/Billing.feature"
},
            glue = {"StepDefinitions"}
)

public class BillingTest {

}
```

Junit-

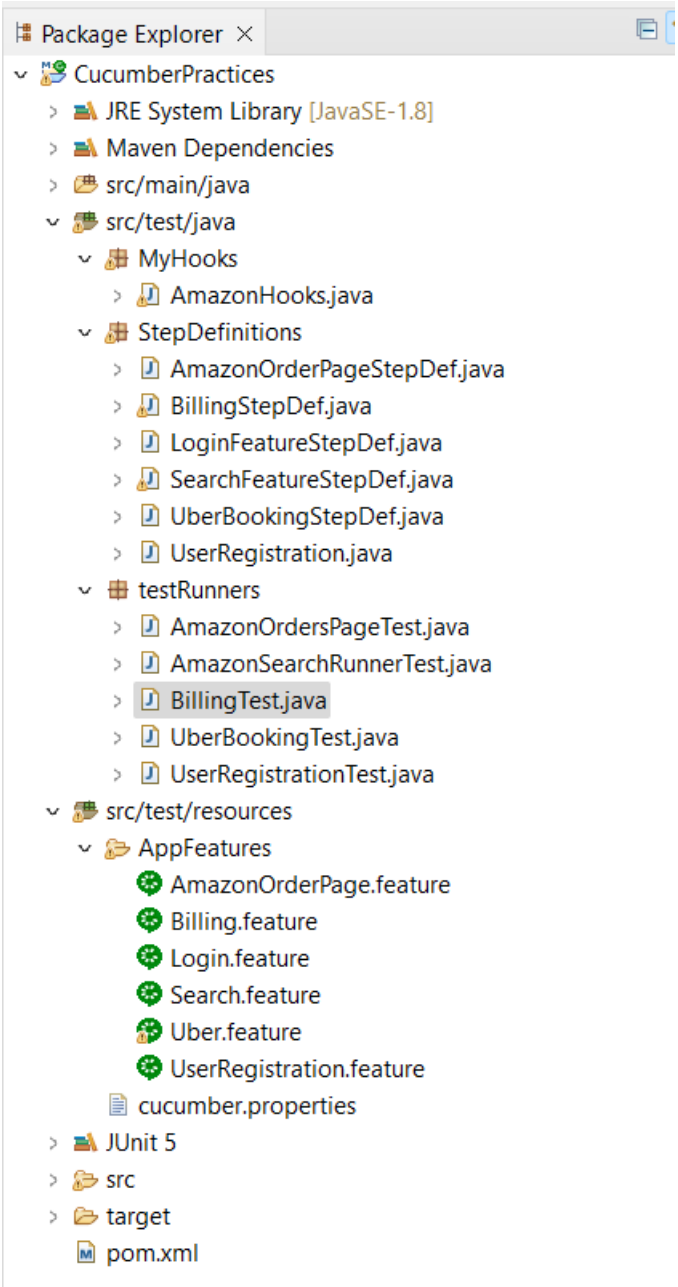


Console-

```
Scenario Outline: bill amount          # src/te
  Given user is on billing page        # StepDe
  When user enters bill amount "1000" # StepDe
  When user enters tax amount "10"     # StepDe
  And user clicks calculate button     # StepDe
expected final amount is  1010.0
Actual final amount is  1010.0
  Then final amount is given "1010"    # StepDe

Scenario Outline: bill amount         # src/tes
  Given user is on billing page       # StepDef
  When user enters bill amount "100" # StepDef
  When user enters tax amount "40"    # StepDef
  And user clicks calculate button    # StepDef
expected final amount is  140.0
Actual final amount is  140.0
  Then final amount is given "140"    # StepDef

Scenario Outline: bill amount        # src/test
  Given user is on billing page      # StepDefi
  When user enters bill amount "20" # StepDefi
  When user enters tax amount "6.7" # StepDefi
  And user clicks calculate button   # StepDefi
expected final amount is  26.7
Actual final amount is  26.7
  Then final amount is given "26.7" # StepDefi
```

We have accomplished data driven testing.

Project structure-



Codes for this lecture-

Feature file for login –

```
Feature: login feature

Scenario Outline: failed login - different combinations
      Given user is on application landing page
      When user clicks on signin button
      Then user is displayed login page
      When user enters "<userName>" in username field
      When user enters "<password>" in password field
      And user clicks on signin button
      Then error displayed for wrong credentials

            Examples:
```

```
                    | userName | password |
                    | incorrectusername | 234324 |
                    | naveen auto | incorrectpassword |
                    | incorrectusername | incorrectpassword |
```

Step def for invalid login-

```
package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class LoginFeatureStepDef {

	@Given("user is on application landing page")
	public void user_is_on_application_landing_page() {

	}

	@When("user clicks on signin button")
	public void user_clicks_on_signin_button() {

	}

	@Then("user is displayed login page")
	public void user_is_displayed_login_page() {

	}

	@When("user enters {string} in username field")
	public void user_enters_in_username_field(String string) {

	}

	@When("user enters {string} in password field")
	public void user_enters_in_password_field(String string) {

	}

	@Then("error displayed for wrong credentials")
	public void error_displayed_for_wrong_credentials() {

	}

}
```

Billing feature-

```
Feature: calculate billing amount

Scenario Outline: bill amount
	Given user is on billing page
	When user enters bill amount "<billAmount>"
	When user enters tax amount "<taxAmount>"
	And user clicks calculate button
	Then final amount is given "<finalAmount>"

		Examples:
			| billAmount | taxAmount | finalAmount |
			| 1000	| 10 | 1010 |
			| 100 | 40 | 140 |
			| 20 | 6.7 | 26.7 |
```

Billing step def-

```
package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import junit.framework.Assert;

public class BillingStepDef {

	double billingAmount;
	double taxAmount;
	double finalAmount;

	@Given("user is on billing page")
	public void user_is_on_billing_page() {

	}

	@When("user enters bill amount {string}")
	public void user_enters_bill_amount(String billingAmount) {
	    this.billingAmount=Double.parseDouble(billingAmount); //convert
string to double and give to the numeric guy/girl
	}

	@When("user enters tax amount {string}")
	public void user_enters_tax_amount(String taxAmount) {
		this.taxAmount=Double.parseDouble(taxAmount);
	}

	@When("user clicks calculate button")
	public void user_clicks_calculate_button() {

	}

	@Then("final amount is given {string}")
	public void final_amount_is_given(String finalAmount) {
	    this.finalAmount=this.billingAmount+this.taxAmount;
//this.finalAmount is already double as we have declared it.
		System.out.println("expected final amount is " + " " +
Double.parseDouble(finalAmount));
		System.out.println("Actual final amount is " + " " +
this.finalAmount);

Assert.assertTrue(this.finalAmount==Double.parseDouble(finalAmount)); //
if the condition is true then assertion should be passed
	}

}
```

Billing runner-

```
package testRunners;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(
		plugin= {"pretty"},
		features = {"src/test/resources/AppFeatures/Billing.feature"
},
```

```
            glue = {"StepDefinitions"}
)

public class BillingTest {

}
```