TRICENTIS

Test Design Specialist Level 1

STUDENT EXERCISE WORKBOOK

SURVEY

Tricentis Design Specialist | Level 1
Student Exercise Workbook

- Designed to be used with Tricentis Tosca version 12.x

Student Exercise Workbook

This exercise workbook is designed to provide a collection of exercises on the methods and concepts covered in the Tricentis Test Design Specialist Level 1 training.

Legal Notice

Tricentis GmbH
Leonard-Bernstein-Straße 10
1220 Vienna
Austria
Tel.: +43 (1) 263 24 09
Fax: +43 (1) 263 24 09-15
Email: academy@tricentis.com

TABLE OF CONTENTS

PREFACE

About this workbook

This workbook is specifically designed to supplement training of the **Tricentis Test Design Specialist Level 1**.

The workbook is arranged into sections. Each section contains a number of exercises which give detailed instructions on how to perform certain functions in Tricentis Tosca.

Tricentis recommends completing the exercises before continuing to the next section and taking the related online exams in order to achieve high impact learning.

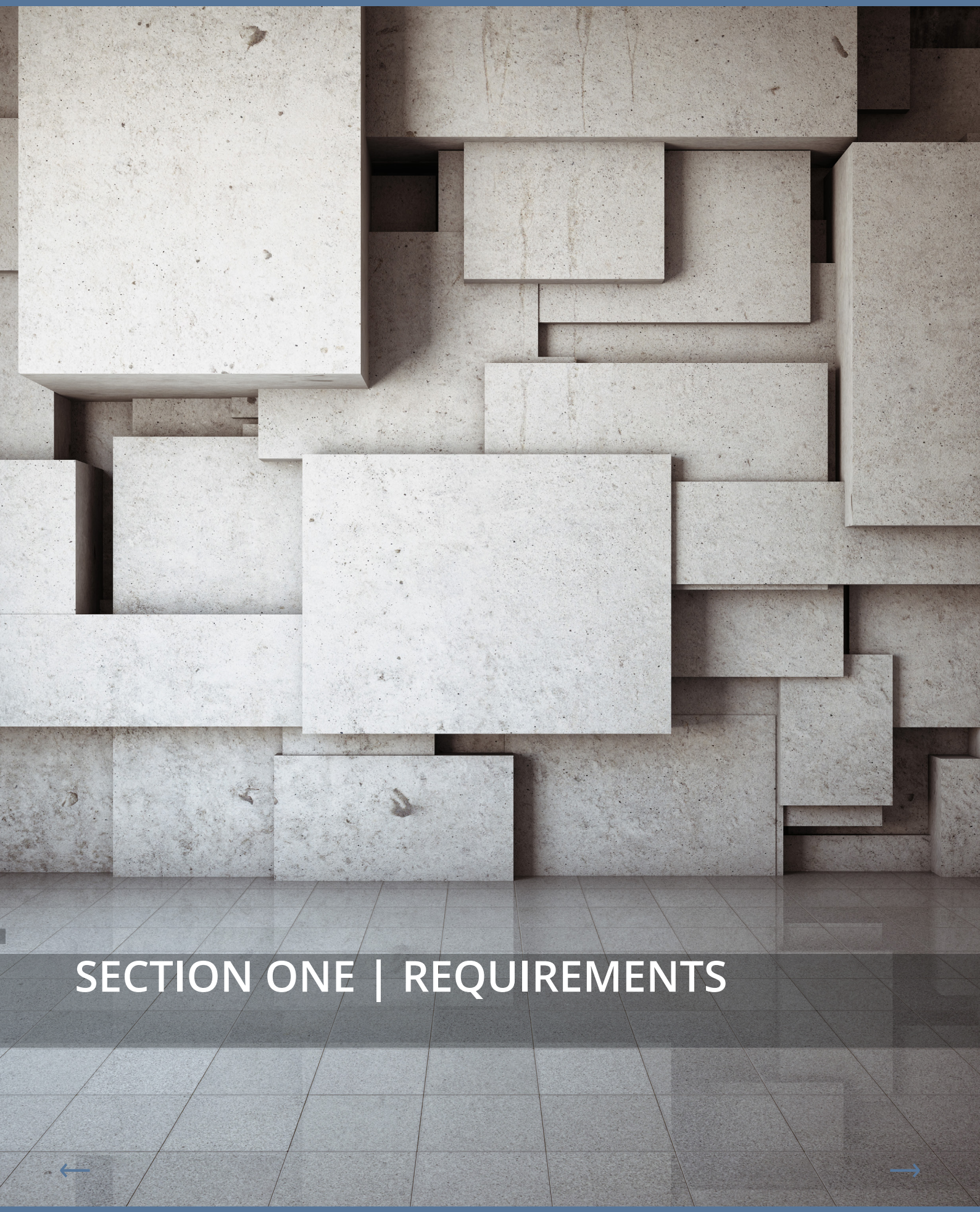
For each exercise, there will be a lesson video that explains how to complete the exercise. Most exercises will also have a solution video that walks through how to complete the exercise in full.

This workbook is not aiming to be a complete manual.

Recommended learning material

In addition to this workbook, it is necessary to use the following material to complete the exercises successfully.
Sample Web Shop application

The Sample Web Shop application is used for most of the exercises. Please use the following link to start the Sample Web Shop application: http://demowebshop.tricentis.com/



SECTION ONE | REQUIREMENTS

REQUIREMENTS

Exercise 01a | Create a Basic Structure

Objective

By the end of this exercise, you will be able to create a basic Requirement structure for the DemoWebShop.

Why is this important?

These are the first steps in building your Requirement structure.

Key elements:

Requirement Set | Requirement | Requirement Folder

Instructions

1.	Within the Requirements Folder "**Requirements**", create a Folder named "**Backlog**".	☐
2.	Within the Requirements Folder "**Backlog**", create a Folder named "**Exercise 1a Create a Basic Structure**".	☐
3.	Within the Requirements Folder "**Exercise 1a Create a Basic Structure**", create a Requirement Set named "**Demo Web Shop**".	☐
4.	Within the Requirement "**Demo Web Shop**", create two Requirements: • **Customer Tasks** • **Handle Products**	☐
5.	Within the Requirement "**Customer Tasks**", create the following Sub-Requirements: • **Register** • **Log In** • **Modify Customer Data** • **Check Orders**	☐
6.	Within the Requirement "**Handle Products**", create the following Sub-Requirements: • **Product Configuration** • **Modify Products View** • **Compare Products** • **Search for Products**	☐

Hints

» Requirements can be created using "Create Object" in the Ribbon, "Create Requirement" in the Mini Toolbar or the shortcut "Ctrl+N Ctrl+R" which creates a Sub-Requirement.

» "Create Requirement (after this)" in the Mini Toolbar or the shortcut "Ctrl+," creates a Requirement on the same level as the one currently selected.

Exercise 01b | Basic Weighting

Objective

By the end of this exercise, you will be able to add a basic weighting structure to the basic Requirement structure.

Why is this important?

Weighting is at the heart of a risk based testing project. This exercise will allow you to practice the first steps in weighting your project.

Instructions

1.	Please enable the "**AutoCalculateRequirements**" setting: Navigate to *Project>>Settings>>Commander>>General>>Advanced*. For the setting "**AutoCalculateRequirements**", selecting "**On**" will enable the auto update function. **Tosca must be restarted to confirm this setting change**.	☐
2.	Duplicate the Folder: "**Exercise 1a Create a Basic Structure**" and rename it: "**Exercise 1b Basic Weighting**".	☐
3.	Expand the Requirement Set "**Demo Web Shop**". Ensure that the column "**Weight**" is shown. If not, use the "**Column Chooser**" to add it.	☐
4.	For the Requirement "**Customer Tasks**", add the Weight **4**. For the Requirement "**Handle Products**", add the Weight **3**.	☐
5.	Add the Weights for the Sub-Requirements as per the table below:	☐

Requirement	Sub-Requirement	Weight
Customer Tasks	Register	3
	Log In	5
	Modify Customer Data	2
	Check Orders	2
Handle Products	Product Configuration	4
	Modify Products View	3
	Compare Products	1
	Search for Products	3

Exercise 01c | Expand the Structure

Objective

By the end of this exercise, you will be able to expand the Requirement structure to reflect the Requirement Set "Front End" for the Demo Web Shop

Why is this important?

This will allow you to create a more realistic Requirement structure.

Instructions

1. Duplicate the Folder: "**Exercise 1b Basic Weighting**" and rename it: "**Exercise 1c Expand the Structure**". ☐
2. Add the remaining **Requirements** into the Requirement Set "**Demo Web Shop**" as per the table below: ☐

Requirement	Sub-Requirement
Shopping Cart	**Add Products**
	Gift Cards
	Discounts
	Manage Shopping Cart
Order Process	**Execute Checkout**
	Billing and Shipping Address
	Calculate Shipping Cost
	Payment Methods
	Re-Order

Hints

» Shortcuts can make creating a Requirement structure much quicker.

» You can use the "Create Requirement structure from Clipboard" functionality to create multiple Requirements at once if you copy all names into the Clipboard.

Exercise 01d | Weighting with Frequency & Damage

Objective

By the end of this exercise, you will be able to weight the Requirements using the Frequency and Damage classes.

Why is this important?

Using Frequency and Damage classes to weight the Requirements is a much more accurate method to calculate risk.

Instructions

1. Duplicate the Folder: "**Exercise 1c Expand the Structure**" and rename it: "**Exercise 1d Weighting with Frequency & Damage**". ☐
2. Expand the Requirement "**Customer Tasks**". Ensure that the columns "**Frequency class**" and "**Damage class**" are shown. If not, use the "**Column Chooser**" to add them. ☐
3. Within the Requirement "**Customer Tasks**", add the following Values: ☐
   - **Frequency class: 4**
   - **Damage class: 4**
4. Add the remaining Frequency and Damage Values for the main Requirements as per the table below: ☐

Requirement	Frequency class	Damage class
Handle Products	3	3
Shopping Cart	5	5
Order Process	5	5

5. Add the Values for the Frequency and Damage for the Sub-Requirements as per the table below: ☐

Requirement	Sub-Requirement	Frequency class	Damage class
Customer Tasks	Register	3	4
	Log In	5	4
	Modify Customer Data	3	2
	Check Orders	2	3
Handle Products	Product Configuration	3	3
	Modify Products View	4	2
	Compare Products	1	1
	Search for Products	4	2
Shopping Cart	Add Products	3	5
	Gift Cards	3	4
	Discounts	3	4
	Manage Shopping Cart	3	4
Order Process	Execute Checkout	3	5
	Billing and Shipping Address	2	5
	Calculate Shipping Costs	5	5
	Payment Methods	5	5
	Re-Order	2	4

Exercise 01e | Structure Sprint #1

Objective

By the end of this exercise, you will be able to create a Requirement structure for a Sprint within an Agile working environment.

Why is this important?

Learning how to use the Requirements structure within different working methodologies is important.

Instructions

1.	Navigate to the Requirements Folder named "**Requirements**". Within this Folder, create a new Folder named "**Sprint Backlogs**".	☐
2.	Within the Folder "**Sprint Backlogs**", create a Folder named: "**Exercise 1e Structure Sprint #1**".	☐
3.	Within the Folder "**Exercise 1e Structure Sprint #1**", create a Requirement Set named "**Sprint #1**".	☐
4.	Within the Requirement Set, create the following Requirements: • **"US1: As a user, I want to order different products, ship them using different methods and pay the correct shipping fee"** • **"US2: As a user, I want to order use different payment methods and pay the correct payment fee"** • **"US3: As a user, I want to use discount codes and have them applied correctly"** • **"US4: As a user, I want to adapt products myself and have the changes be reflected accordingly"**	☐

Hints

» If necessary, further information can be added to the "Description Column".

Exercise 01f | Weighting Sprint #1

Objective

By the end of this exercise, you will be able to use Frequency and Damages classes to weight the Requirements.

Why is this important?

This will allow you to see that the weighting for Sprints is maintained in the same manner.

Instructions

1. Duplicate the Folder: "**Exercise 1e Structure Sprint #1**" and rename it: "**Exercise 1f Weighting Sprint #1**". ☐
2. Weight the Requirements according to the table below: ☐

Requirement	Frequency	Damage
US1: As a user, I want to order different products, ship them using different methods and pay the correct shipping fee	**5**	**5**
US2: As a user, I want to order use different payment methods and pay the correct payment fee	**5**	**4**
US3: As a user, I want to use discount codes and have them applied correctly	**2**	**4**
US4: As a user, I want to adapt products myself and have the changes be reflected accordingly	**2**	**3**

SECTION TWO | TEST CASE DESIGN

TestSheet 1 | Calculate Shipping Costs Overview

Business Context "Calculate Shipping Costs"

The foundation of online shopping is the shipping of the ordered goods to the customer. In theory, for every order that needs to be shipped, a certain amount of shipping cost applies. The key question to ask, is: who pays for the shipping, vendor or customer?

In our DemoWebShop, we need to test if the shipping costs are correctly calculated. The business department informed us about the following detailed requirements of this functionality:

Product types:

- There are two different types of products sold:
  - ◊ Physical products that need to be shipped to the customer – shipping costs do apply – this is the standard variant
  - ◊ Digital products that can be downloaded directly – shipping costs do not apply

Order size:

- Customers can purchase in different order sizes and combinations:
  - ◊ A single physical product – this is the standard variant
  - ◊ Multiple physical products
  - ◊ A single digital product
  - ◊ Multiple digital products
  - ◊ Multiple products of different product types

Shipping methods:

- Customers can use different shipping methods:
  - ◊ Ground, the standard method which costs USD 10.00
  - ◊ Next day air, the most expensive and fastest method, costing USD 40.00
  - ◊ Second day air, slightly slower and less expensive, USD 20.00
  - ◊ Pick it up in one of our stores, naturally free of cost

Shipping costs are paid by:

- In general, the shipping costs are paid by the customer, except:
  - ◊ When there are no shipping costs
  - ◊ When the customers sub-total of all products in the cart exceeds USD 50.00

During the course of this training we will not use all features from the start but introduce some of them at later stages to show additional functionalities within Tosca TestCaseDesign.

TEST CASE DESIGN

Exercise 02a | TestSheet and Basic Attribute Structure

Objective

By the end of this exercise, you will be able to create a TestSheet and the recommended Attribute structure to be used in all TestSheets.

Why is this important?

The recommended base Attribute structure helps you and the testing team to organize and have a single template for all TestSheets, ensuring readability at all levels.

Key elements:

TestSheet | Attribute | Attribute Not Business Relevant | Attribute Result

Instructions

1.	Navigate to the TestCaseDesign section. Within the TestCaseDesign folder, create a new folder named: "**TestSheet 1 Calculate Shipping Costs**".	☐
2.	Within the folder "TestSheet 1 Calculate Shipping Cost", create a sub-folder named: "**Exercise 2a TestSheet and Basic Attribute Structure**".	☐
3.	Within Folder "Exercise 2a TestSheet and Basic Attribute Structure", create a TestSheet named "**Calculate Shipping Costs**".	☐
4.	Click on the TestSheet in the navigation pane and within the TestCaseDesign Ribbon, use the "**Create Object**" button to create an Attribute. Name this Attribute "**Administration**".	☐
5.	Create 3 more Attributes named: • **Precondition** • **Process** • **Verification**	☐
6.	Select your first Attribute "**Administration**" and within the Ribbon use the Toggle Business Relevance button to change the Property **BusinessRelevant** to "**No**".	☐
7.	In the Attribute "**Verification**", change the Property **BusinessRelevant** to "**Result**".	☐
8.	Add the following Sub-Attributes according to the table below:	☐

Attribute	Sub-Attribute
Administration	**Test Designer**
	Contact Person (Business)
	Test Stage
	Comment

Hints

- Attributes can be created in three ways: 1) using the Ribbon “Create Object”, 2) using the Mini Toolbar “Create Attribute” or 3) using the shortcut “Ctrl+N Ctrl+A”.
- Use the Mini Toolbar and “Create Attribute" (after this) shortcut “Ctrl+,” which creates an Attribute on the same level as the one currently selected.
- The colour of the Attribute icon changes according to the Business Relevance: “No” is pink and “Result” is green.

Exercise 02b | Attribute Structure – Process and Verification

Objective

By end of this exercise, you will be able to add the rest of the Attributes that will make up the TestSheet.

Why is this important?

The Attribute structure should be built up in a logical manner, this exercise will demonstrate this.

Instructions

1. Duplicate the folder: "Exercise 2a TestSheet and Basic Attribute Structure" and rename it: "**Exercise 2b Attribute Structure – Process and Verification**". ☐
2. Add the following Attributes to the TestSheet according to the table below: ☐

Attribute	Sub-Attribute
Precondition	**Customer**
Process	**Ordered Products**
	Checkout
Verification	**Message**
	Order Details

3. Add the next level of Attributes, by adding further Sub-Attributes to the just created Sub-Attributes according to the table below (see next page for full table overview): ☐

Precondition

Customer

- Type of User
- Address

Process

Ordered Products

- Product
  - ◊ Product Type
  - ◊ Article
  - ◊ Quantity

Checkout

- Billing Address
- Shipping Address
  - ◊ Shipping Address
  - ◊ In-Store Pickup
- Payment Method
  - ◊ Credit Card Information
    - ¤ Card Type
    - ¤ Card Number
    - ¤ Expiration Date
    - ¤ Card Code

Verification

Message

Order Details

- Shipping Cost
- Total Price

Hints

» Tricentis Best Practice is to have a maximum of 10-15 Requirements for each level for maintainability, comparability and a good overview.

Exercise 03a | Instance Structure – Administration and Precondition

Objective

By the end of this exercise, you will be able to create the Instance structure for the Attributes "Administration" and "Prerequisite".

Why is this important?

Creating a logical and consistent Instance structure is vital for a successful TestSheet.

Key elements:

Instance

Instructions

1. Duplicate the Folder: "**Exercise 2b Attribute Structure – Process and Verification**" and rename it "**Exercise 3a Instance Structure – Administration and Precondition**". ☐

Context

From now on we will introduce the **Context** section. This information is additional to the exercise steps and will give context to the instructions in the exercises.

The Instances for the Administration Attribute should now be added. Remember that these instances supply greater information regarding the setup of the Test Sheet. It will not affect the number or make up of the TestCases.

2. Focus on the Attribute "**Test Designer**" within the Attribute "**Administration**". Use the Ribbon to add the Instances:
   - **Max Methodology**
   - **Steven Design**

   ☐
3. Within each Sub-Attribute, add the **Instances** as per the table below: ☐

Sub Attribute	Instances
Contact Person	**Peter Business**
	Carl Business
Test Stage	**Regression**
	Smoke
	One Time

4. Focus on the Sub-Attribute "**Customer**" (*Precondition>>Customer)*. Add the **Instance** to the Sub-Attributes as per the table below: ☐

Sub Attribute	Instances
Type of User	**Registered**
Address	**Available**

Hints

- Instances can be created in three ways: 1) using the button in the Ribbon “Create Object”, 2) using the Mini Toolbar “Create Attribute” or 3) shortcut “Ctrl+N Ctrl+I” which creates a Sub-Attribute.
- Use the Mini Toolbar and “Create Instance (after this) shortcut “Ctrl+,” which creates an Instance on the same level as the one currently selected.

Exercise 03b | Instance Structure – Process and Verification

Objective

By the end of this exercise, you will be able to create the Instance structure for the Attributes "Process" and "Verification".

Why is this important?

This will walk you through the process of creating a more complex Instance structure.

Instructions

1. Duplicate the Folder: "**Exercise 3a Instance Structure – Administration and Precondition**" and rename it "**Exercise 3b Instance Structure – Process and Verification**". ☐

Context

We now need to create the Instances within the Attributes "Process" and "Verification". In the Attribute "Product", we need to add the types of product that will be required for our TestCases, not the exact products themselves.

When purchasing products in the WebShop, shipping costs depend on the ordered product (e.g. for digital downloads no shipping costs apply) and the value of the order's sub-total (with an order sub-total of more than $50, no shipping costs apply).

To reflect this, we need the following categories:

- Physical product that will need to be shipped e.g a pair of Blue Jeans
- A digitally downloaded product that requires no shipping e.g. 3rd Album
- Product or number of products that would lead to free shipping

The boundary values for the shipping costs also now need to be considered. So, two more Instances are required:

- The boundary for free shipping
- Boundary where shipping costs apply

2. Create the Instance structure as per the table below: ☐

Attribute	Attribute Path	Instances
Product	*Process>>Ordered Products>>Product*	**Physical**
		Digital
		Free Shipping
		Free Shipping | Boundary
		Shipping Costs Applied | Boundary

Context

The rest of the Instances are more straightforward. There are three Attributes relating to the addresses:

- Billing address will always revert to the "Default" address in the system so the default address is the only one needed.
- Shipping address: we need two Instances, one where the shipping address is the same as the billing address, and another where no shipping is required for when a digital product is ordered.
- In-Store Pickup: Three instances, true and false for a physical product that is or is not picked up in-store and N/A for a digital product that cannot be picked up.

Payment methods are not the focus of the tests in this TestSheet. So, within Payment Method, we only add details of the single payment method that we are going to use - "Credit card". Therefore, we need to add an Instance for each Attribute: to describe the type of card and the information the SUT needs to process a payment:

- Type of card: Visa
- Visa card number
- Expiry date
- 3-digit card code

3. Create the Instance structure as per the table below:

Attribute	Attribute Path	Instances
Billing Address	*Process>>Checkout>>Billing Address*	**Default**
Shipping Address	*Process>>Checkout>>Shipping Address>>Shipping Address*	**Equals Billing Address**
		N/A
In-Store Pickup	*Process>>Checkout>>Shipping Address>>In-Store Pickup*	**True**
		False
		N/A
Card Type	*Process>>Checkout>>Payment Method>>Credit Card Information>>Card Type*	**Visa**
Card Number	*Process>>Checkout>>Payment Method>>Credit Card Information >>Card Number*	**4916637380170906**
Expiration Date	*Process>>Checkout>>Payment Method>>Credit Card Information>>Expiration Date*	**Expires in two years**
Card Code	*Process>>Checkout >>Payment Method>>Credit Card Information>>Card Code*	**123**

Context

Finally, the Verification Attribute requires Instances. These are the elements that we will be verifying are correct within the TestCase:

- Attribute "Message" needs an Instance that confirms that the order successful message has appeared.
- "Shipping cost" Attribute: Remembering the business context from earlier in the TestCase, there are three possible options here:
  - ◊ The default shipping costs are applied
  - ◊ Free shipping
  - ◊ No shipping costs
- The Attribute "Total price" is the total value of the order. There are two possibilities:
  - ◊ The order value plus any shipping costs
  - ◊ Just the order value when there are no shipping costs

4. Create the Instance structure as per the table below:

Attribute	Attribute Path	Instances
Message	*Verification>>Message*	**Success Message**
Shipping Cost	*Verification>>Order Details>>Shipping Cost*	**Default Shipping Costs**
		Free Shipping
		No Shipping
Total Price	*Verification>>Order Details>>Total Price*	**Order Value + Shipping Costs**
		Order Value

Exercise 03c | Instance Structure – Set Character and Position

Objective

By the end of this exercise, you will be able to set the Instance Character and Position of the Instances created in the TestSheet.

Why is this important?

Setting the Instance Character and Position helps keep the TestSheet easy to read and makes defining the TestCases considerably simpler.

Key elements:

Valid | Invalid | Straight-Through | Inner value | Boundary value

Instructions

1. Duplicate the Folder: "**3b Instance Structure – Process and Verification**" and rename it "**Exercise 3c Instance Structure - Set Character and Position**". ☐

Context

The StraightThrough Instances need to be defined. For this TestSheet, the business department has advised us that the StraightThrough is:

- Customer address is the default address in the customer's account
- A physical product is ordered
- Billing address is the default address in the customer's account
- There is no in-store pickup
- Visa is the chosen card, with the relevant Visa card Instances

2. Navigate to the Instance "**Registered**" (*Precondition>>Customer>>Type of User*) use the Ribbon button "**Toggle Character**" to change the Character to "**StraightThrough**". ☐
3. Add the "**Character**" to the rest of the Instances as per the table on the following page: ☐

Instance	Path	Character
Available	*Precondition>>Customer>>Address>>Available*	**StraightThrough**
Physical	*Process>>Ordered Products>>Product>>Physical*	**StraightThrough**
Default	*Process>>Checkout>>Billing Address>>Default*	**StraightThrough**
Equals Billing Address	*Process>>Checkout>>Shipping Address>>Shipping Address>>Equals Billing Address*	**StraightThrough**
False	*Process>>Checkout>>Shipping Address>>In-Store Pickup>> False*	**StraightThrough**
Visa	*Process>>Checkout>>Payment Method>>Credit Card Information>>Card Type>>Visa*	**StraightThrough**
4916637380170906	*Process>>Checkout >> Payment Method>>Credit Card Information>>Card Number>>4916637380170906*	**StraightThrough**
Expires in two years	*Process>>Checkout>>Payment Method>>Credit Card Information>>Expiration Date>>Expires in two years*	**StraightThrough**
123	*Process>>Checkout>>Payment Method>>Credit Card Information>>Card Code>>123*	**StraightThrough**

Context

The Boundary Instances now need to be identified and marked as such. These can be identified by looking back at the business context for Exercise 3b.

4. Identify the Boundary Values in the table below and change their "**Position**" to "**Boundary**" using the "**Toggle Position**" Button in the Ribbon. ☐

Instance	Path	Position
Free Shipping | Boundary	*Process>>Ordered Products>>Product>>Free Shipping | Boundary*	**Boundary**
Shipping Costs Applied | Boundary	*Process>>Ordered Products>>Product>>Shipping Costs Applied | Boundary*	**Boundary**

Hints

- Instances Characters and position can also be changed in the Properties tab of the Instance.
- A StraightThrough cannot have its Position changed, as a StraightThrough must always be "Inner" and "Valid".
- There can only be one StraightThrough in Tosca per "Instance Group".
- The Character "Invalid" is not used in this TestSheet but can be set in the same way as "Straight Through" and "Valid".
- The Shortcut to change the Character is F7.
- The Shortcut to change the Position is F8.

Exercise 04a | Data Combinations - Manual Linear Expansion - Lower Levels

Objective

By the end of this exercise, you will be able to combine the Attributes to create Instances at the lower levels of the TestSheet.

Why is this important?

Combining these Attributes reflects the actual data combinations needed to test the Requirement.

Instructions

1. Duplicate the Folder: "**Exercise 3c Instance Structure - Set Character and Position**" and rename it "**Exercise 4a Data Combinations - Manual Linear Expansion - Lower Levels"**. ☐

Context

The lower level Attributes now need to be combined. To start, the Attribute "Customer" needs to be combined. For this TestSheet, there is only one type of "Customer", which would be a Registered User with an address that is already available in the system. This customer type will be the StraightThrough.

2. Navigate to the Attribute "**Customer**" which is a Sub-Attribute of "**Precondition**". Right click on the Attribute "**Customer**" and select "**Create Instance**". Rename this Instance "**Registered | Address Available**". ☐
3. Navigate to where the "**Registered | Address Available**" Instance Column intersects with the "**Type of user**" Attribute Row. From the drop-down box select the Value "**Registered**". ☐
4. Navigate to where the "**Registered | Address Available**" Instance Column intersects with the "**Address**" Attribute Row. From the drop-down box select the Value "**Available**". ☐
5. Expand the Instance Folder underneath the Attribute "**Customer**", right click on the Instance "**Registered | Address Available**", in the Ribbon use the Toggle Character button to set this Instance as the "**StraightThrough**". ☐

Context

The Instances within the Attribute "Products" have been created in a previous exercise. This means it is now possible to move up a level to the "Ordered Products" Attribute and combine. As "Product" is the only Sub-Attribute within the "Order Products" Attribute, there is no real combination needed, so the Instances within the Sub-Attribute "Product", including their Characters and Position can be duplicated at the higher level.

Values for the Sub-Attribute "Product" can also be added to fit the flow of the "Ordered Product" Instance.

6. Focus on the Attribute "**Ordered Product**". Then, where the Attribute "**Product**" and the following **Instances** intersect, add the following Values: ☐

Attribute	Instances	Position/Character
Ordered Products	**Physical**	**StraightThrough**
	Digital	**Inner**
	Free Shipping	**Inner**
	Free Shipping | Boundary	**Boundary**
	Shipping Costs Applied | Boundary	**Boundary**

7. Focus on the Attribute "**Ordered Product**". Then, where the Attribute "**Product**" and the following Instances intersect, add the following Values: ☐

	Instances				
	Physical	**Digital**	**Free Shipping**	**Free Shipping | Boundary**	**Shipping Costs Applied | Boundary**
Attributes					
Product	**Physical**	**Digital**	**Free Shipping**	**Free Shipping | Boundary**	**Shipping Costs Applied | Boundary**

Context

The exercise now focuses onto the "Checkout" Attribute. As before the Sub-Attributes should be combined first, then the main "Checkout" Attribute.

There is only one specific credit card being used in this TestSheet as testing the various payment methods is not the focus of this TestSheet. Therefore, in the "Credit Card Information" Sub-Attribute, there is only one Instance:

- Visa Card
  - ◊ The StraightThrough Instances that you created for the Sub-Attributes can all be selected as Values for this card (for example the type of card and the card number).

"Payment Method" only requires one Instance to be created to reflect the single card we are using to pay in this TestSheet. This will be the StraightThrough:

- Credit Card | Visa

Shipping address is more complex as all the types of shipping need to be taken into account and added as Instances:

- The order requires delivery, and the shipping address is the same as the billing address.
- The order does not require delivery because the physical products will be picked up in the store, therefore no shipping address is required.
- Delivery is not required because the order does not have any physical products, for example a digital download which may simply be downloaded online.
  - ◊ Each Instance needs the relevant Values for the Sub-Attribute selected.

The Attribute "Billing Address" has already had the one Instance needed created in an earlier exercise.

8. Focus on the Attribute "**Credit Card Information**" (Process>>Checkout>>Payment Method>>Credit Card Information), add the Instance "**Visa Card**" with the "**StraightThrough**" Character. Then, where the Instance "**Visa Card**" and the following Attributes intersect, add the following Values: ☐

	Instance
	Visa Card
Attributes	
Card Type	**Visa**
Card Number	**4916637380170906**
Expiration Date	**Expires in two years**
Card Code	**123**

9. Focus on the Attribute "**Payment Method**" (Process>>Checkout>>Payment Method), add the Instance "**Credit Card | Visa**" with the "**StraightThrough**" character. Then, where the Instance "**Credit Card | Visa**" and the "**Credit Card**" Attribute intersect, add the following Value: ☐

	Instances
	Visa Card
Attributes	
Credit Card Information	**Visa Card**

10. Focus on the Attribute "**Shipping Address**" (Process>>Checkout>>Shipping Address). Create the Instances:
- **Equals Billing Address" (StraightThrough)**
- **"In-Store Pickup" (valid inner)**
- **"N/A" (valid inner)**

Where these Instances intersect with the following Attributes, add the following Values: ☐

	Instances		
	Equals Billing Address	In-Store Pickup	N/A
Attributes			
Shipping Address	**Equals Billing Address**	**N/A**	**N/A**
In-Store Pickup	**False**	**True**	**N/A**

Context

The next level now needs to be combined, which is the Attribute "Checkout". This means creating Instances to the Checkout Attribute and assigning Values for Billing Address, Shipping Address, and Payment Method. Since there are multiple Instances for these Attributes, our first Instance will be the StraightThrough which combines all StraightThrough Instances for these Attributes. Then there is one additional Instance combination for each non-StraightThrough Instance. There are two non-StraightThrough Instances, so in total there are 3 combinations:

- Standard (StraightThrough): standard checkout process is when an order is placed using the default billing address, delivery is required and is sent to the same shipping address as the billing address, and the payment method is a Visa credit card.
- In-Store Pickup: this Instance should be the same as the StraightThrough but we switch out one non-StraightThrough Instance. In this case, shipping is not required so the option for in-store pickup is chosen instead.
- No shipping: all Values stay the same as the StraightThrough, except the Instance "Shipping Address" which is changed to "Not Applicable".

11. Focus on the Attribute "**Checkout**" (Process>>Checkout). Create the following Instances including their Character and Position:
- **"Standard" (StraightThrough)**
- **"In-Store Pickup" (valid inner)**
- **"No Shipping" (valid inner)**

Where these Instances intersect with the following Attributes, add the following Values: ☐

	Instances		
	Standard	In-Store Pickup	N/A
Attributes			
Billing Address	**Default**	**Default**	**Default**
Shipping Address	**Equals Billing Address**	**In-Store Pickup**	**N/A**
Payment Method	**Credit Card | Visa**	**Credit Card | Visa**	**Credit Card | Visa**

Exercise 04b | Data Combinations - Manual Linear Expansion - TestSheet Level

Objective

By the end of this exercise you will be able to combine the attributes to create instance at the TestSheet Level.

Why is this important?

Combining these Attributes creates the Instances that relate to the actual TestCases that need to be created.

Instructions

1. Duplicate the folder: "**Exercise 4a Data Combinations - Manual Linear Expansion - Lower Levels**" and rename it "**Exercise 4b Data Combinations - Manual Linear Expansion - TestSheet Level**". ☐

Context

The Attributes need to be combined at the TestSheet level. The Attributes to be combined are:

- Customer - Only one StraightThrough Instance
- Ordered Products - 5 Instances
- Checkout - 3 instances

For each of these 3 Attributes, one of the Instances is defined as a StraightThrough and all others are non-StraightThrough Instances. Logically using Linear Expansion, this will result in 7 Instances. However, we have a dependency that if a Digital product is ordered, there will be No Shipping. This combines two of the Instances to create 6. The following Instances should be created:

- ◊ Purchase of a Physical product (StraightThrough): a customer places an order to their registered address. They purchase a physical product and use the standard checkout process.
- ◊ Purchase of a digital product: a customer places an order to their registered address. They purchase a digital product (non-StraightThrough) and therefore no shipping applies.
- ◊ Free shipping: a customer uses their registered address to make a purchase of physical goods worth over $51, so shipping costs are free, and use the standard checkout process.
- ◊ In-Store Pickup: a customer uses their registered address to make a purchase of physical goods which will be picked up in-store, so no shipping costs should be incurred.
- ◊ Free shipping boundary Value: a customer uses their registered address to make a purchase of physical goods worth exactly $51, which is the boundary at which no shipping costs apply.
- ◊ Shipping cost applied boundary Value: a customer uses their registered address to make a purchase of physical goods worth exactly $50, which is the boundary at which shipping costs do apply.

This may appear to be less Instances than you would expect. This is because Using Linear Expansion and StraightThrough, some Instances are duplicated and are not required. For example there is no Instance for Standard Delivery, as it would be identical to the Instance that is created for the purchase of a physical product.

2. Focus at the TestSheet level, expand the Attributes:
   - **Precondition**
   - **Process**
   ☐
3. Right click on the TestSheet "**Calculate Shipping Costs**" and click the button "**Create Instance**" from the Mini Toolbar. Name this Instance "**Physical**". ☐
4. Select the following Values from the dropdown boxes in the yellow boxes next to the Attributes according to the table below: ☐

Attribute	Value
Customer	**Registered | Address Available**
Ordered Products	**Physical**
Checkout	**Standard**

5. Set the Character of the Instance to "**StraightThrough**". ☐

6. Create the Instances including the Character and Position:
   - **"Free Shipping | Boundary" (valid Boundary)**
   - **"Digital" (valid inner)**
   - **"Free Shipping" (valid inner)**
   - **"In-Store Pickup" (valid inner)**
   - **"Shipping Costs Applied | Boundary" (valid Boundary)**

   Where these Instances intersect with the following Attributes, add the following Values: ☐

	Instances				
	Free Shipping | Boundary	Digital	Free Shipping	In-Store Pickup	Shipping Costs Applied | Boundary
Attributes					
Shopping Cart	**Registered | Address Available**	**Registered | Address Available**	**Registered | Address Available**	**Registered | Address Available**	**Registered | Address Available**
Ordered Products	**Free Shipping | Boundary**	**Digital**	**Free Shipping**	**Physical**	**Shipping Costs Applied | Boundary**
Checkout	**Standard**	**No Shipping**	**Standard**	**In-Store Pickup**	**Standard**

7. Arrange the Instances by selecting the TestSheet and choosing "**Arrange Instances**" from the "**Optimize Instances**" button in the Ribbon. ☐

Hints

» If you want to fill in Values all at once you can make use of the function "Fill empty Values" and then "Specific" from the context menu on the respective Attribute.

Exercise 05 | Values

Objective

By the end of this exercise, you will be able to add the Values to the TestSheet to further specify the future TestCases.

Why is this important?

The more detailed the information for the Automation Specialists, the quicker they will be able to work and the less errors should occur.

Instructions

1. Duplicate the Folder: "**Exercise 4b Data Combinations - Manual Linear Expansion - TestSheet Level**" and rename it "**Exercise 5 Values**". ☐

Context

The Values that will be used when the TestCases are created can now be added to the TestSheet. The Administration Attribute requires some basic admin information. However, this will not be required in the TestCase creation phase.

Other Values may require Attributes to be created. For example, in the "Type of User" Attribute, the StraightThrough Instance is that the User is "Registered". However, here the actual Values for "User Name" and "Password" that should be used can be specified. As all combinations have already taken place, there is no effect on the number of TestSheet Instances.

2. Focus on the TestSheet level, and expand the Attribute "**Administration**". Add the following Values for the Attributes for **all** the TestCase Instances: ☐
   - **Test Designer:** **Max Methodology**
   - **Contact Person (Business):** **Peter Business**
3. Add the following Values for the Attribute "**Test Stage**" according to the table below: ☐

	Instances					
	Physical	**Digital**	**Free Shipping**	**In-Store Pickup**	**Free Shipping | Boundary**	**Shipping Costs Applied | Boundary**
Attributes						
Test Stage	**Smoke**	**Regression**	**Regression**	**Regression**	**One Time**	**One Time**

4. Focus on the Attribute "**Type of User**" (*Precondition>>Customer>>Type of User*) and add two further Attributes: ☐
   - **Username**
   - **Password**
5. In the newly created Attributes, enter the Values as per the table below: ☐

Attribute	Value
Username	**email@company.com**
Password	**12345678**

Context

The Values for the Product Attribute need to be added now. In this case, the products chosen should meet the Requirements of each individual TestSheet Instance. For example, a physical product (Blue Jeans) should be chosen. The choice and quantity of the product must take into account the total price, as this effects whether shipping costs are applicable or not, and is necessary to calculate the correct Boundary Values (shipping costs apply on an order of a physical product of $50 or below).

6. Focus on the "**Product**" Attribute (*Process>>Ordered Products>>Product*).
Where the Attributes intersect with the following TestCase Instances, add the Values according to the table below: ☐

	Instances				
	Physical	**Digital**	**Free Shipping**	**Free Shipping | Boundary**	**Shipping Costs Applied | Boundary**
Attributes					
Product Type	**Physical**	**Digital**	**Physical**	**Physical**	**Physical**
Article	**Blue Jeans**	**3rd Album**	**Blue Jeans**	**Blue Jeans**	**Blue Jeans**
Quantity	**1**	**1**	**70**	**51**	**50**

Context

The Instances and Values for the credit card's expiration date can now be added. In this case, a dynamic expression will be used as the Value. Dynamic expressions can also be used in TestSheets and linked to TestSheet Templates. As you will remember from Automation Specialist Level 1, this dynamic date expression will ensure that the expiry date is always 2 years in the future.

7. Focus on the "**Expiration Date"** Attribute (*Process>>Checkout>>Payment Method>>Credit Card Information>>Expiration Date*). Add the Attributes: ☐
   - **Month**
   - **Year**

   Add the Values as per the table below:

Attribute	Value
Month	**{DATE[][][MM]}**
Year	**{DATE[][+2y][yyyy]}**

Context

The Verification Attribute is now populated with data. This data will tell the Automation Specialist the actual Values to be verified within this TestSheet. This is two areas: The success message, that is, that the SUT tells the user that an order was placed successfully. Secondly, the order details, which is the financial information broken down as it appears in the SUT, so the shipping cost, and the total price will both have the expected Values to be verified entered.

8. Focus on the "**Message**" Attribute (*Verification>>Message*). Add the Attribute:
   - **Value**

   Add the Value as per the table below: ☐

	Instances
	Success Message
Attributes	
Value	**Your order has been successfully processed**

9. Focus on the TestSheet and navigate to and expand the "**Verification**" Attribute. Use the Dropdown box to add the Value "**Success Message**" to the "**Message**" Attribute where it intersects with all Instances at the TestSheet level. ☐

10. Focus on and expand the "**Shipping Cost**" Attribute (*Verification>>Order Details>>Shipping Cost*). Add the Attribute:
    - **Value**

    Where the Attribute "**Value**" intersects with the following TestSheet Instances, add the Values according to the table below: ☐

	Instances		
	Default Shipping Costs	**Free Shipping**	**No Shipping**
Attributes			
Value	**10.00**	**0.00**	**Not Required**

11. Focus on and expand the Attribute "**Order Details**" (*Verification>>Order Details*). Add the Attribute:
    - **Value Total Price** ☐

12. Focus on the TestSheet. Where the Attribute "**Value Total Price**" intersects with the following TestCase Instances, add the Values according to the table below: ☐

	Instances					
	Physical	**Digital**	**Free Shipping**	**In-Store Pickup**	**Free Shipping | Boundary**	**Shipping Costs Applied | Boundary**
Attributes						
Value Total Price	**11.00**	**1.00**	**70.00**	**1.00**	**51.00**	**60.00**

13. Focus on the TestSheet. Where the Attributes "**Shipping Cost**" and "**Total Price**" intersect with the following TestSheet Instances, add the Values from the drop down boxes according to the table below: ☐

	Instances					
	Physical	Digital	Free Shipping	In-Store Pickup	Free Shipping | Boundary	Shipping Costs Applied | Boundary
Attributes						
Shipping Costs	**Default Shipping Costs**	**No Shipping**	**Free Shipping**	**No Shipping**	**Free Shipping**	**Default Shipping Costs**
Total Price	**Order Value + Shipping Costs**	**Order Value**	**Order Value + Shipping Costs**	**Order Value**	**Order Value + Shipping Costs**	**Order Value + Shipping Costs**

Hints

» Using “Fill Empty Values” and the Filter function can make entering data into the TestSheet much quicker.

Exercise 06a | Create and Adapt Class

Objective

By the end of this exercise, you will be able to create and adapt a class for the Attribute "Product".

Why is this important?

This exercise demonstrates how classes can be created and adapted.

Key elements:

Class

Class Reference

Instructions

1.	Duplicate the Folder: "**Exercise 5 Values**" and rename it "**Exercise 6a Create and Adapt Class**".	☐
2.	In the "**TestCaseDesign**" Folder, create a Folder named "**Classes**".	☐
3	Within the Folder named "**Classes**" create a Sub-Folder named "**Exercise 6a Create and Adapt Class**".	☐
4.	Navigate to the Folder "**Exercise 6a Create and Adapt Class**" within the "**TestSheet 1 Calculate Shipping Costs**" Folder and focus on the Attribute "**Product**".	☐
5.	Drag the Attribute "**Product**" and drop it in to the "**Classes**" within the Folder "**Exercise 6a Create and Adapt Class**" to create a new **Class** named "**Product**".	☐

Context

Having created a Class, we now need to adapt it to the changing circumstances of our SUT. The ordering of a second product should now be included in the "Products" Attribute. The Instances that need to be created are:

- A second physical product
- A high quantity of digital Items
- N/A for when there isn't a second product

These Instances will then allow us to deepen the testing of the shipping costs.

6.	Navigate to the Class named "**Product**" and add the following 3 Instances: • **"Physical | 2nd Product" (Inner/Valid)** • **"Digital | High Quantity" (Inner/Valid)** • **"N/A" (Inner/Valid)**	☐
7.	Focus on the Class "**Product**". Add the **Values** for the 3 new Instances as per the table below (there should be no Values entered for the Instance "N/A"):	☐

	Instances		
	Physical | 2nd Product	Digital | High Quantity	N/A
Attributes			
Product Type	**Physical**	**Digital**	
Article	**Casual Golf Belt**	**3rd Album**	
Quantity	**1**	**70**	

8. Use “**Arrange Instances**” to reorganize the Instances within the **Class** “**Product**”. ☐

Hints

» After dragging and dropping the Attribute to create a class, the symbol on the Attribute in the TestSheet changes and will have a white arrow on a red background indicating the Class Reference.

» To get a page overview, you can deselect *Options* > *TestCaseDesignUI* > *Hide EmbeddedScrollArea navigation columns* and in the line above select the number of columns to be displayed.

Exercise 06b | Use Class and Complete Instances

Objective

By the end of this exercise, you will be able to add the additional TestSheet Instances at the lower level required due to the addition of new products.

Why is this important?

If a TestSheet is amended, the impact to the combinations must be taken into account.

Instructions

1.	Duplicate the Folder: "**Exercise 6a Create and Adapt Class**" which contains the TestSheet and rename it "**Exercise 6b Use Class and Complete Instances**".	☐
2.	Navigate and expand the "**Ordered Products**" Attribute. Navigate to and expand the Folder "**Classes**" in the Folder "**Exercise 6a Create and Adapt Class**". Drag the Class "**Product**" and drop it into the "**Ordered Products**" Attribute.	☐
3.	Rename Attribute "**Product**" to "**1st Product**". Rename the newly created Attribute "**Product_1**" to "**2nd Product**".	☐

Context

Three new combinations must be added to the "Ordered Products" Level. This is to test three new areas:

- Shipping costs are not added twice when two products are ordered.
- If a high quantity of digital products is added to an order containing a physical product that would normally incur shipping costs, no shipping costs are incurred as the order exceeds the $50 threshold.
- An order that includes a physical product and a digital product but has a total value below $50 still incurs shipping costs.

4.	Focus on the "**Ordered Product**" Attribute. Add 3 TestCase Instances: • **2 Physical (Inner/Valid)** • **Physical + Digital | Shipping Cost (Inner/Valid)** • **Physical + Digital | Free Shipping (Inner/Valid)**	☐
5.	Arrange the Instances by selecting the Attribute "**Ordered Products**" and chose "**Arrange Instances**" from the "**Optimize Instances**" button in the Ribbon. Make sure the final order of the Instances appear as follows: • **Physical (StraightThrough)** • **Digital** • **2 Physical (Inner/Valid)** • **Physical + Digital | Shipping Cost (Inner/Valid)** • **Physical + Digital | Free Shipping (Inner/Valid)** • **Free Shipping (Inner/Valid)** • **Free Shiping | Boundary (Inner/Invalid)** • **Shipping Costs Applied | Boundary (Inner/Invalid)**	☐

6. Navigate to and expand the "**Ordered Products**" Attribute (*Process>>Ordered Products*). Where the Attributes intersect with the following TestSheet Instances, add the Values according to the table below: ☐

	Instances							
	Physical	Digital	2 Physical	Physical + Digital | Shipping Cost	Physical + Digital | Free Shipping	Free Shipping	Free Shipping | Boundary	Shipping Costs Applied | Boundary
Attributes								
1st Product	**Physical**	**Digital**	**Physical**	**Physical**	**Physical**	**Free Shipping**	**Free Shipping | Boundary**	**Shipping Costs Applied | Boundary**
2nd Product	**N/A**	**N/A**	**Physical | 2nd Product**	**Digital**	**Digital | High Quantity**	**N/A**	**N/A**	**N/A**

Exercise 06c | Complete TestSheet Level Instances

Objective

By the end of this exercise, you will be able to create and complete the necessary TestCase Instances.

Why is this important?

The relevant TestCase Instances must now be created at the TestSheet level and, in order for all the TestCases to be readable for Automation Specialists, Values must be entered.

Instructions

1. Duplicate the Folder: "**Exercise 6b Use Class and Complete Instances**" and rename it "**Exercise 6c Complete TestSheet Level Instances**". ☐
2. On the TestSheet level, add the 3 new Instances: ☐
   - **2 Physical (Inner/Valid)**
   - **Physical + Digital | Shipping Cost (Inner/Valid)**
   - **Physical + Digital | Free Shipping (Inner/Valid)**
3. Complete the TestSheet Instances entering the Values where they are required as per the table below: ☐

	Instances		
	2 Physical	**Physical + Digital | Shipping Cost**	**Physical + Digital | Free Shipping**
Attributes			
Administration:			
Test Designer	**Max Methodology**	**Max Methodology**	**Max Methodology**
Contact Person	**Peter Business**	**Peter Business**	**Peter Business**
Test Stage	**Regression**	**Regression**	**Regression**
Precondition:			
Customer	**Registered | Address Available**	**Registered | Address Available**	**Registered | Address Available**
Process:			
Ordered Products	**2 Physical**	**Physical + Digital | Shipping Cost**	**Physical + Digital | Free Shipping**
Checkout	**Standard**	**Standard**	**Standard**
Verification:			
Message	**Success Message**	**Success Message**	**Success Message**
Order Details / Shipping Cost	**Default Shipping Costs**	**Default Shipping Costs**	**Free Shipping**
Order Details/ Total Price	**Order Value + Shipping Costs**	**Order Value + Shipping Costs**	**Order Value + Shipping Costs**

4. At the TestSheet level, focus on the Attribute "**Value Total Price**". Add the Values that correspond to the actual Values that will be verified according to the table below: ☐

	Instances		
	2 Physical	**Physical + Digital | Shipping Cost**	**Physical + Digital | Free Shipping**
Attributes			
Value Total Price	**12.00**	**12.00**	**71.00**

Exercise 07 | Link the Values

Objective

By the end of this exercise, you will be able to link the completed TestSheet to the correct Requirements.

Why is this important?

Linking the TestSheet now will give an overview of what TestCases need to be created, and will also provide a general overview of the entire project later on.

Key elements:

TestCase Link

TestCase Link Subtitute

Instructions

1.	Navigate to the **Requirements** section. Duplicate the Folder: "**Exercise 1d Weighting with Frequency & Damage**" and rename it "**Final Linked Requirements**".	☐
2.	Duplicate the Folder: "**Exercise 1f Weighting Sprint #1**" and rename it "**Final Linked Requirements**".	☐
3.	Open the **TestCaseDesign** section and open the Folder "**Exercise 6c Complete Test Sheet Level Instances**". Drag the TestSheet "**Calculate Shipping Costs**" and drop it on to the Requirement "**Calculate Shipping Costs**" (*Requirements>>Backlog>>Final Linked Requirements>>Demo Web Shop>>Order Process>>Calculate Shipping Costs*).	☐
4.	Drag the same TestSheet "**Calculate Shipping Costs**" and drop it onto the Requirement "**US1: As a user, I want to order different products, ship them using different methods and pay the correct shipping fee**" (*Requirements>>Sprint Backlogs>>Final Linked Requirements>>Sprint #1>>US1: As a user, I want to order different products, ship them using different methods and pay the correct shipping fee*)	☐

Exercise 08 | Integrate New Attribute

Objective

By the end of this exercise, you will be able to add a new Attribute to the TestSheet and fully integrate it at all necessary levels.

Why is this important?

As the SUT develops, it may become necessary to add Attributes that will affect the overall number of TestSheet Instances, these need to be correctly integrated into the TestSheet.

Instructions

1.	Duplicate the Folder: "**Exercise 6c Complete Test Sheet Level Instances**" and rename it "**Exercise 8 Integrate New Attribute**".	☐

Context

New Attributes need to be added to take into account all the different shipping methods and different costs applied:

- Ground Shipping (the StraightThrough): $10.00
- Next Day Air: $40.00
- Second Day Air: $20.00
- A fourth "Not Applicable" Instance which can be used when shipping methods are not applicable

On the Checkout level, new Values are required for existing Instances, and additional Instances will need to be created for the new shipping methods: "Next Day Air" and "Second Day Air". These Instances should be populated with the StraightThrough Values except for the "Shipping Method" Attribute which will have the relevant Value chosen.

The other level that requires amendment is the TestSheet Level. Two new TestSheet Instances should be added for the two methods of air delivery. Again, they should all have StraightThrough Instances except for the "Checkout" Attribute where the correct Attributes "Next Day Air" and "Second Day Air" should be selected.

A TestSheet Instance for N/A is not required as this is used in the "Digital" and "In-Store Pickup" TestSheet Instances.

2.	Navigate to the Attribute "**Checkout**" (*Process>>Checkout*). Add the Attribute "**Shipping Method**" and add the following Instances: • **Ground (Straight Through)** • **Next Day Air (Inner/Valid)** • **Second Day Air (Inner/Valid)** • **N/A (Inner/Valid)**	☐
3.	In the Attribute "**Checkout**", add the two new Instances: • **Next Day Air** • **Second Day Air**	☐
4.	For the Instance: "**Standard**", add the **StraightThrough** Values to the Instances where they intersect the Attribute "**Shipping Method**". For the Instances "**In-Store Pickup**" and "**No Shipping**", add the Value "**N/A**" where they intersect the Attribute "**Shipping Method**".	☐
5.	For the Instances "**Next Day Air**" and "**Second Day Air**", add the **StraightThrough** Values where they intersect for all Attributes except "**Shipping Method**".	☐
6.	For the Instances "**Next Day Air**", select the Value "**Next Day Air**", and for the Instance "**Second Day Air**" select the Value "**Second Day Air**" where the Attribute "**Shipping Method**" intersects the Instance.	☐
7.	Navigate to the "**Shipping Cost**" Attribute (*Verification>>Order Details>>Shipping Costs*) and add the two new Instances: • **Next Day Air** • **Second Day Air**	☐

8. For the Instances "**Next Day Air**" and "**Second Day Air**", enter the following Values where the "**Value**" Attribute intersects the new Instances according to the table below: ☐

	Instances	
	Physical	Digital
Attributes		
Shipping Cost		
Value	**40.00**	**20.00**

9. Focus on the TestSheet Level. Add the TestSheet Instances:
   - **Next Day Air**
   - **Second Day Air**

   Add the Values where the Instances intersect the Attributes as per the table below: ☐

	Instances	
	Next Day Air	Second Day Air
Attributes		
Administration		
Test Designer	**Max Methodology**	**Max Methodology**
Contact Person	**Peter Business**	**Peter Business**
Test Stage	**Regression**	**Regression**
Precondition		
Customer	**Registered | Address Available**	**Registered | Address Available**
Process		
Ordered Products	**Physical**	**Physical**
Checkout	**Next Day Air**	**Second Day Air**
Verification		
Message	**Success Message**	**Success Message**
Order Details / Shipping Cost	**Next Day Air**	**Second Day Air**
Order Details / Total Price	**Order Value + Shipping Costs**	**Order Value + Shipping Costs**
Order Details / Value Total Price	**41.00**	**21.00**

Context

Having created new Instances, these must now be linked to the Requirements. In our case, because we are duplicating the TestSheets prior to amending them, we have two separate TestSheets with links. In a real life scenario, this would not be the case as we would amend one TestSheet and not duplicate it and amend it so there are two "live" TestSheets.

10. Link the two new Instances: "**Next Day Air**" and "**Second Day Air**" to the relevant **Requirement** (see Ex.07). ☐

Hints

- "Create Instance (after this)" (Ctrl+,) creates an empty Instance on the same level. "Duplicate (after this)" (Ctrl+.) duplicates all aspects so that all the Values are already populated.
- The easiest way of manually adding an Instance (e.g. for a Linear Expansion), is to duplicate (Ctrl+.) the StraightThrough Instance, change the name and Character of the Instance and alter the Value of the Attribute which needs to be adjusted.

Exercise 09a | Create and Adapt Class

Objective

By the end of this exercise, you will be able to create and adapt Classes from the now complete TestSheet 1.

Why is this important?

Classes greatly reduce the amount of work required to create a new TestSheet. They can also be adapted to allow a greater variety of uses in other TestSheets.

Instructions

1. In the "**Classes**" Folder, create a Sub-Folder named: "**Exercise 9a + 10c Create and Adapt Classes**". ☐

Context

Now, we need to start thinking about our next TestSheet "Calculate Payment Fee". The aim of this TestSheet is to check that any additional charges are only levied when the payment methods "Cash on Delivery" or "Check / Money Order" are used.

Many of the elements from TestSheet 1 can be used in this new TestSheet. Therefore, Classes can be created from these reusable elements.

After creating the Classes, it will have to be taken into account that in the previous TestSheet, payment methods were not the test focus. Therefore, only the StraightThrough Instance for Visa Card was necessary. Now, however, the other types of payment methods need to be added to the Class:

- Credit Card – Master Card
- Credit Card – Discover
- Credit Card – American Express
- Purchase Order
- Cash on Delivery

The Class will then need to have the appropriate combinations added as well as the appropriate Values.

The additions to the Class will not affect the original TestSheet as the StraightThrough Instance has already been selected. Unused Instances in a TestSheet are not a problem.

As well as the Values for the different Credit Cards for which Attributes and Instances already exist, an additional Attribute with relevant Instances will need to be created for the Purchase Order number.

2. Navigate to the TestSheet Folder "**Exercise 8 Integrate New Attribute**". Navigate to the "**Customer**" Attribute (*Precondition>>Customer*). Drag and drop the "**Customer**" Attribute into the "**Exercise 9a + 10c Create and Adapt Classes**" Class Folder to create a Class. ☐

3. Navigate to and expand the "**Checkout**" Attribute. Drag the following Attributes into the "**Exercise 9a + 10c Create and Adapt Classes**" Class Folder to create Classes: ☐
   - Billing Address
   - Shipping Address
   - Shipping Method
   - Payment Method

4. Navigate to the Class "**Payment Method**". Add the Attribute: "**Purchase Order Number**". Add the Instances: ☐
   - N/A (StraightThrough)
   - 123456 (Valid Inner)

5.	Navigate to the Attribute "**Credit Card Information**" (*Payment Method>>Credit Card Information*) and add the Instances: • Master Card (Valid Inner) • Discover (Valid Inner) • American Express (Valid Inner) • N/A (Valid Inner)	☐
6.	Navigate to the Attribute "**Card Type**" (*Payment Method>>Card Type*) and add the Instances: • Master Card (Valid Inner) • Discover (Valid Inner) • American Express (Valid Inner)	☐
7.	Navigate to the Attribute "**Card Number**" (*Payment Method>>Card Number*) and add the Instances: • 5468560510822353 (Valid Inner) • 6011550865535692 (Valid Inner) • 372115991558863 (Valid Inner)	☐
8.	Focus on the Class "**Payment Method**" and add the following Instances: • Credit Card | Master Card (Valid Inner) • Credit Card | Discover (Valid Inner) • Credit Card | American Express (Valid Inner) • Purchase Order (Valid Inner) • Cash on Delivery (Valid Inner)	☐
9.	Focus on the Attribute "**Credit Card Information**". Add the Values from the drop-down boxes where the Instances intersect the Attributes as per the table below:	☐

	Instances			
Attributes	**Credit Card | Master Card**	**Credit Card | Discover**	**Credit Card | American Express**	**N/A**
Credit Card Information:				
Card Type	**Master Card**	**Discover**	**American Express**	
Card Number	**5468560510822353**	**6011550865535692**	**372115991558863**	
Expiration Date	**Expires in two years**	**Expires in two years**	**Expires in two years**	
Card Code	**123**	**123**	**123**	

10.	Focus on the Class "**Payment Method**". Add the Values from the drop-down boxes where the Instances intersect the Attributes as per the table below:	☐

	Instances				
Attributes	**Credit Card | Master Card**	**Credit Card | Discover**	**Credit Card | American Express**	**Purchase Order**	**Cash on Delivery**
Payment Method:					
Credit Card Information	**Master Card**	**Discover**	**American Express**	**N/A**	**N/A**
Purchase Order Number	**N/A**	**N/A**	**N/A**	**123456**	**N/A**

TestSheet 2 | Calculate Payment Fee Overview

Business Context "Calculation of Payment Fees"

Another important part of online business is getting the money for the goods that are sold. There are, therefore, a multitude of different payment methods provided, some free of charge, some with a charge applied.

In our DemoWebShop, we need to test if the payment fees are correctly applied. The business department has supplied the following detailed Requirements of this functionality:

Payment methods

There are four payment methods available for our customers:

Credit Card – this is the standard variant

- Purchase by Credit Card is free of any fees for the customer
- 4 different Credit Card types may be used
    - ◊ Visa – this is the standard variant
    - ◊ Master Card
    - ◊ Discover
    - ◊ American Express

Purchase Order

- This is free of any fees for the customer
- A purchase order number is required

Cash on Delivery

- A fee of USD 7.00 applies

Check / Money Order

- A fee of USD 5.00 applies

Please note, Lesson 10 will further demonstrate how these different payment fees are applied at the different levels within the TestSheet.

Exercise 09b | Use Classes

Objective

By the end of this exercise, you will be able to create a TestSheet from the modified Classes.

Why is this important?

Classes help to create TestSheets quickly and efficiently.

Instructions

1. Within the TestCaseDesign Folder "**TDS1_TestCaseDesign**", create a new Folder named "**TestSheet 2 Calculate Payment Fee - Tool supported Creation of Instances**". ☐
2. Within the TestCaseDesign Folder named "**TestSheet 2 Calculate Payment Fee - Tool supported Creation of Instances**", create a Folder named "**Exercise 9b Build Structure - Use Classes**". ☐
3. Within the Folder "**Exercise 9b Build Structure - Use Classes**", create a TestSheet named "**Calculate Payment Fee**". ☐

Context

The basic structure of this TestSheet is very similar to the "Calculate Shipping Costs" one. There is one main difference however: in the first TestSheet, the type of product ordered directly influenced the type of shipping costs applied. Therefore, the product choice needed to be included in the "Process" Attribute. In this TestSheet however, the product ordered is not relevant; a product simply needs to be ordered to incur a payment. Therefore, the "Product" Attribute is a "Precondition" step and not a "Process" step. The only "Process" steps are the "Checkout" Attributes.

4. Create the **Attribute** structure for the TestSheet "**Calculate Payment Fee**" as per the table below: ☐

Administration (Business Relevant – No)

- Test Designer
- Contact Person (Business)
- Test Stage
- Comment

Precondition

- Class "Customer"
- Class "Product"

Process

- Checkout
  - ◊ Class: "Billing Address"
  - ◊ Class: "Shipping Address"
  - ◊ Class: "Shipping Method"
  - ◊ Class: "Payment Method"

Verification (Business Relevant – Result)

- Message
- Order Details
  - ◊ Shipping Cost
  - ◊ Total Price

4. Create the Attribute structure for the TestSheet "**Calculate Payment Fee**" as per the table below: ☐

Sub-Attribute	Instances
Test Designer	**Max Methdology**
	Steven Design
Contact Person	**Peter Business**
	Carl Business
Test Stage	**Regression**
	Smoke
	One Time

Exercise 10a | Combine to Checkout - Combine Precondition

Objective

By the end of this exercise, you will be able to automatically combine Instances.

Why is this important?

Tricentis Tosca offers the ability to automatically generate Instances using Linear Expansion.

Instructions

1.	Duplicate the Folder: "**Exercise 9b Build Structure - Use Classes**" and rename it "**Exercise 10a Combine to Checkout - Combine Precondition**".	☐

Context

As with manual linear expansion, the process starts at the lower Attribute levels and then moves upwards. Therefore, the expansion starts at the Attributes: "Precondition" and "Checkout". It is important to decide what Attributes require combining.
For the Attribute "Precondition", it is only the type of customer as the type of product is not important. This will be the standard (StraightThrough) purchase.
For the Attribute "Checkout", the focus of the TestSheet is on the payment methods. Therefore, only the "Payment Methods" Attribute requires combination.

2.	Focus on the Attribute "**Precondition**". Create an Instance, but then immediately **delete** the Instance that will have the default name "**Precondition**" to leave just the Instance Folder.	☐
3.	Select the Attribute "**Customer**". Right click and select "**Generate Instances > Linear Expansion**".	☐
4.	Focus on the "**Precondition**" Attribute. Rename the StraightThrough Instance "**Standard**". Where "**Product**" intersects the Instance "**Standard**", add the Value "**Physical**" from the drop-down box.	☐
5.	Select the Attribute "**Checkout**". Create an Instance, but then immediately **delete** the Instance that will have the default name "**Checkout**" to leave just the Instance Folder.	☐
6.	Select the Attribute "**Payment Method**". Right click and select "**Generate Instances > Linear Expansion**".	☐
7.	For all Values that have not been populated, select the **StraightThrough** Instance.	☐

Exercise 10b | Combine to TestSheet and add Values

Objective

By the end of this exercise, you will be able to automatically combine Instances at the TestSheet level.

Why is this important?

This demonstrates how to add the TestSheet Instances automatically using Linear Expansion.

Instructions

1.	Duplicate the Folder: "**Exercise 10a Combine to Checkout - Combine Precondition**" and rename it "**Exercise 10b Combine to TestSheet and add Values**".	☐

Context

As before, the Instances need to be completed at the TestSheet Level. Tosca can do this automatically. Values for the "Verification" Attribute should also be added. As before, the success message will be verified.

In the "Order Details" Verification there are two options: 1) the order Value: when a payment method that does not incur a charge is used (e.g. Credit Card), and 2) the order Value plus the payment fee: when a payment method that incurs a fee is used (e.g. Cash on delivery).

For the Value of the total price, a dynamic expression is being used. This will tell the Automation Specialist to Buffer the price of the Blue Jeans and then multiply it by the number of products to be ordered in the TestSheet.

2.	Navigate to the "**Calculate Payment Fee**" TestSheet level. Right click on the TestSheet and select "**Generate Instances > Linear Expansion**".	☐
3.	Navigate to the "**Message**" Attribute (*Verification>>Message*). Add the Instance: • **Success Message**	☐
4.	Within the "**Message**" (*Verification>>Message*) Attribute, add an Attribute named "**Value**". Add the Instance: • **Your order has been successfully processed** Where the Attribute "**Value**" intersects with the Instance "**Success Message**", select the Value "**Your order has been successfully processed**".	☐
5.	Navigate to the "**Payment Fee**" Attribute (*Verification>>Order Details>>Payment Fee*), and add the Instances: • **No Fee** • **Cash on Delivery Fee**	☐
6.	Within the "**Payment Fee**" Attribute (*Verification>>Order Details>>Payment Fee*), add an Attribute named "**Values**".	☐
7.	Navigate to the "**Total Price**" Attribute (*Verification>>Order Details>>Total Price*), and add the Instances: • **Order Value** • **Order Value + Payment Fee**	☐
8.	Within the "**Total Price**" Attribute (*Verification>>Order Details>>Total Price*), add an Attribute named "**Value**".	☐
9.	Within the Attribute Named "**Value**", add the Instances: • **{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product.Quantity]}]}** • **{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product.Quantity]}+{XL[Verification.Order Details. Payment Fee]}]}**	☐
10.	Enter the appropriate Values for the TestSheet as per the table below:	☐

	Instances					
	StraightThrough	Credit Card |Master Cart	Credit Card |Discover	Credit Card |American Express	Purchase Order	Cash on Delivery
Attributes						
Administration						
Test Designer	**Steven Design**	**Steven Design**	**Steven Design**	**Steven Design**	**Steven Design**	**Steven Design**
Contact Person	**Carl Business**	**Carl Business**	**Carl Business**	**Carl Business**	**Carl Business**	**Carl Business**
Test Stage	**Smoke**	**Regression**	**Regression**	**Regression**	**Regression**	**Regression**
Precondition						
Precondition	**Standard**	**Standard**	**Standard**	**Standard**	**Standard**	**Standard**
Process						
Checkout	**Standard**	**Credit Card | Master Card**	**Credit Card | Discover**	**Credit Card | American Express**	**Purchase Order**	**Cash on Delivery**
Verification						
Message	**Success Message**	**Success Message**	**Success Message**	**Success Message**	**Success Message**	**Success Message**
Order Details / Payment Fee	**No Fee**	**No Fee**	**No Fee**	**No Fee**	**No Fee**	**Cash on Delivery Fee**
Order Details / Payment Fee / Values	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**	**7.00**
Order Details/ Total Price	**Order Value**	**Order Value**	**Order Value**	**Order Value**	**Order Value**	**Order Value + Payment Fee**
Order Details/ Total Price / Value	***See table below***					

Instance	Value
StraightThrough	**{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product.Quantity]}]}**
Credit Card | Master Card	**{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product.Quantity]}]}**
Credit Card | Discover	**{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product.Quantity]}]}**
Credit Card | American Express	**{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product.Quantity]}]}**
Purchase Order	**{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product.Quantity]}]}**
Cash on Delivery	**{MATH[{B[PriceBlueJeans]}*{XL[Precondition.Product. Quantity]}+{XL[Verification.Order Details.Payment Fee]}]}**

11. Link the TestSheet to the relevant Requirements: "**Payment Methods**" (*Backlog>>Final Linked Requirements>>Demo Web Shop>>Order Process>>Payment Methods*) and "**US2: As a user, I want to order use different payment methods and pay the correct payment fee**" (*Sprint Backlogs>>Final Linked Requirements>>Sprint#>>US2: As a user, I want to order use different payment methods and pay the correct payment fee*). ☐

Hints

» Instead of creating all Attributes from scratch, they can be copied from a previous TestSheet and pasted into the new one. Bear in mind that amendments may be necessary.

Exercise 10c | Combine to TestSheet and add Values

Objective

By the end of this exercise, you will be able to amend the TestSheet and complete the Instance structure.

Why is this important?

As before, systems change and Test Sheets need to change too. This exercise demonstrates how the addition of a new Attribute is easily accommodated.

Instructions

1. Duplicate the Folder: "**Exercise 10b Combine to TestSheet and add Values**" and rename it "**Exercise 10c Adapt Class and Complete Instances, Adaptations --> Final TestSheet**". ☐

Context

There has been an additional function added to the SUT, that a customer can pay for the goods that have been ordered using a check or Money Order. There is an additional charge for this service of $5.00.
This additional payment service must be accommodated in the TestSheet, with the necessary change to the class / TestSheet.

2. Navigate to the Classes Folder "**Exercise 9a + 10c Create and Adapt Classes**". Navigate to the "**Payment Method**" Class and add the Instance "**Check / Money Order**". ☐
3. Add the Values to the Instance "**Check / Money Order**" as per the table below: ☐

	Instances
	Check / Money Order
Class	
Payment Method	
Credit Card Information	**N/A**
Purchase Order Number	**N/A**

4. Navigate to the "**Calculate Payment Fee**" TestSheet in Folder "**Exercise 10c Adapt Class and Complete Instances, Adaptations --> Final TestSheet**". Focus on the "**Checkout**" Attribute. Right click on the Attribute "**Payment Method**" and select "**Complete Instances > Linear Expansion**". ☐
5. Navigate to the Attribute "**Payment Fee**" (*Verification>>Order Details>>Payment Fee*). Add the Instance "**Check / Money order Fee**". Enter the Value of "**5.00**" where the "**Values**" Attribute intersects with the "**Check / Money order Fee**" Instance. ☐
6. Focus on the TestSheet. Right click on the TestSheet and select "**Complete Instances > Linear Expansion**". ☐
7. Complete the Values for the new TestSheet Instances as per the table below: ☐

	Instances
	Check / Money order Fee
Attributes	
Administration	
Test Designer	**Steven Design**
Contact Person	**Carl Business**
Test Stage	**Regression**
Precondition	
Precondition	**Standard**
Process	
Checkout	**Check / Money Order**
Verification	
Message	**Success Message**
Order Details / Payment Fee	**Check / Money Order**
Order Details/ Total Price	**Order Value + Payment Fee**

8. Link the TestSheet Instance "**Check / Money Order**" to the relevant Requirements: "**Payment Methods**" (*Backlog>>Final Linked Requirements>>Demo Web Shop>>Order Process>>Payment Methods*) and "**US2: As a user, I want to order use different payment methods and pay the correct payment fee**" (*Sprint Backlogs>>Final Linked Requirements>>Sprint#>>US2: As a user, I want to order use different payment methods and pay the correct payment fee*). ☐

SECTION THREE | SCENARIOS

SCENARIOS

Scenario 01 | Create the TestSheet Calculate Discount

Business Context "Calculation of Discounts"

A third part of online business is promotions and discounts where various prices in various amounts apply.

In our DemoWebShop, we need to test if the discount amounts are correctly calculated and applied. The business department informed us about the following detailed requirements of this functionality:

Discount types:

There are two types of discounts:

- **Percentages** – this is the standard variant
- **Flat discounts**

Discount assignment:

There are three options where discounts can be assigned to:

- **Total price** – this is the standard variant
- **Sub-Total price**
- **Shipping costs**

How to use them:

Discounts are assigned using a code that must be valid.

Valid Discount Codes:

- **PercentageSubtotal** (deducting 10%)
- **PercentageTotal** (deducting 10%)
- **PercentageShipping** (deducting 10%)
- **FlatTotal** (deducting USD 5)

Scenario 02 | Create the TestSheet Product Configuration

Business Context "Product Configuration"

For some products, it is very important that customers can configure them the way they want. This can apply to clothing (size, color) or for example, electronic products.

In our DemoWebShop, we need to test if products can be configured. The business department informed us about the following detailed requirements of this functionality:

Product names:

Two products were prepared as test products for testing the configuration:

- **Build your own cheap computer** – this is the standard variant
  - ◊ Base Price is USD 800.00
- **Build your own expensive computer**
  - ◊ Base Price is USD 1,800.00

Processor:

There are three types of processors that can be used:

- **Medium speed** – this is the standard variant. Additional price is USD 15.00
- **Fast speed** – additional price is USD 100.00
- **Slow speed** – no additional charge

RAM:

There are three sizes of RAM that can be used:

- **2 GB** – no additional charge
- **4 GB** – this is the standard variant. Additional price is USD 20.00
- **8 GB** – additional price is USD 60.00

Hard Disk Drive:

There are two sizes of Hard Disk Drives available:

- **320 GB** – this is the standard variant. No additional charge
- **400 GB** – additional price is USD 100.00

Software:

There is additional software available:

- **An Image Viewer** – additional price is USD 5.00
- **An Office Suite** – additional price is USD 100.00
- **Another Office Suite** – additional price is USD 40.00

Multiple software can be combined, but both Offices Suites may not be bought with the same system.

Scenario 03 | Create the TestSheet Accelerator Package

Business Context "Accelerator Packages"

This is an additional, abstract example that is not found in the DemoWebShop. We are creating a TestSheet for a mobile phone package:

Billable Features:

Four features can be billed:

- **Voice calls**
- **SMS**
- **MMS**
- **Data**

Costs:

There are three types of cost applied:

A yearly fee

- USD 20.00
- It is split into monthly payments of USD 1.67

A monthly fee

- USD 15.00

Usage fees

- Voice Calls
  - ◊ 1,500 minutes are included
  - ◊ Each additional minute costs USD 0.35
- SMS
  - ◊ 1,000 SMS are included
  - ◊ Each additional SMS costs USD 0.10
- MMS
  - ◊ Each MMS costs USD 0.35
- Data usage
  - ◊ 2 GB included with up to 2 Mbit/s Download and 1 Mbit/s upload
  - ◊ After 2 GB have been used, all bandwidth is reduced to 64 Kbit/s

Notifications:

Text notifications will be sent:

- If **80%** of units are used up
- For each type of unit (SMS, Voice calls, Data) **separately**