

First Mandatory Assignment

Unit Testing and Integration Testing

We have to develop an online shopping application for the phone company KEANet, according to the following interface:

The screenshot shows a web application window titled "KEANet". Inside the window, there is a checkbox labeled "Internet connection". Below it is a label "Phone lines" followed by a text input field containing the number "0". Underneath is the label "Cell phones:". To the left of this label is a list box containing five items: "Motorola G99", "iPhone 99", "Samsung Galaxy 99", "Sony Xperia 99", and "Huawei 99". To the right of the list box are two buttons, ">" and "<". To the right of these buttons is an empty list box. At the bottom left of the window, it says "Total price: 0 DKK". At the bottom center is a button labeled "Buy".

The user should be able to do the following:

- Hire an internet connection
- Choose the number of phone lines s/he wants to hire
- Buy one or more than one cell phones

The maximum number of phone lines that can be hired is 8.

The "Total price" label at the bottom of the window will be updated in real time as the user selects/deselects/increments/decrements/adds/removes elements.

In case a user wants more than one unit of the same cell phone model, s/he will add it to the list on the right as many times as needed.

When clicking the "Buy" button:

- If there is nothing selected, an alert will pop up suggesting the user to select something.
- Otherwise, the list of selected items will be displayed on an alert.

Price list:

Service	Price	Regularity
Internet connection	200DKK	Monthly
Phone line	150DKK	Monthly
Motorola G99	800DKK	Once
iPhone 99	6000DKK	Once
Samsung Galaxy 99	1000DKK	Once
Sony Xperia 99	900DKK	Once
Huawei 99	900DKK	Once

We will implement a Purchase class with the following characteristics (use your own naming notation):

- Attributes:
 - Internet connection (boolean)
 - Phone lines (integer)
 - Cell phones (array of strings)
 - Price (integer)
- Methods:

Description	Parameters	Return value
Including/excluding the Internet connection	boolean	The new total price
Incrementing the number of phone lines	N/A	The new total price
Decrementing the number of phone lines	N/A	The new total price
Selecting a cell phone	string: model name	The new total price
Unselecting a cell phone	string: model name	The new total price
Buying	N/A	The string to be shown in the alert

Tasks (groups of 3):

- Choose a programming language(s) of your choice to implement the application
- Assign tasks within the group, so that every group member has to program some code
- Create a code repository accessible by all group members (a public github repository can be a good option)
- Create unit tests for all the methods in the Purchase class
- Design test cases to achieve 100% decision coverage using the appropriate black-box techniques. Document your use of black-box techniques
- Implement a data provider to execute all unit tests in a row
- Implement a continuous integration job in the CI/CD tool of your choice that:
 - Integrates all application components
 - Runs all unit tests
 - Generates a deployable artifact

The following deliverables will be uploaded in a zip file to the folder “First Mandatory Assignment” on Fronter:

- The application’s source code and the unit tests, or a link to your Github repository containing both (in this case, the contents of the repository must not change after the delivery deadline)
- The documentation regarding test case design
- Proof of success for the CI/CD process (configuration files and job output)