**Homework: Class Diagram, ObjectAid (hw\_cdoa)**

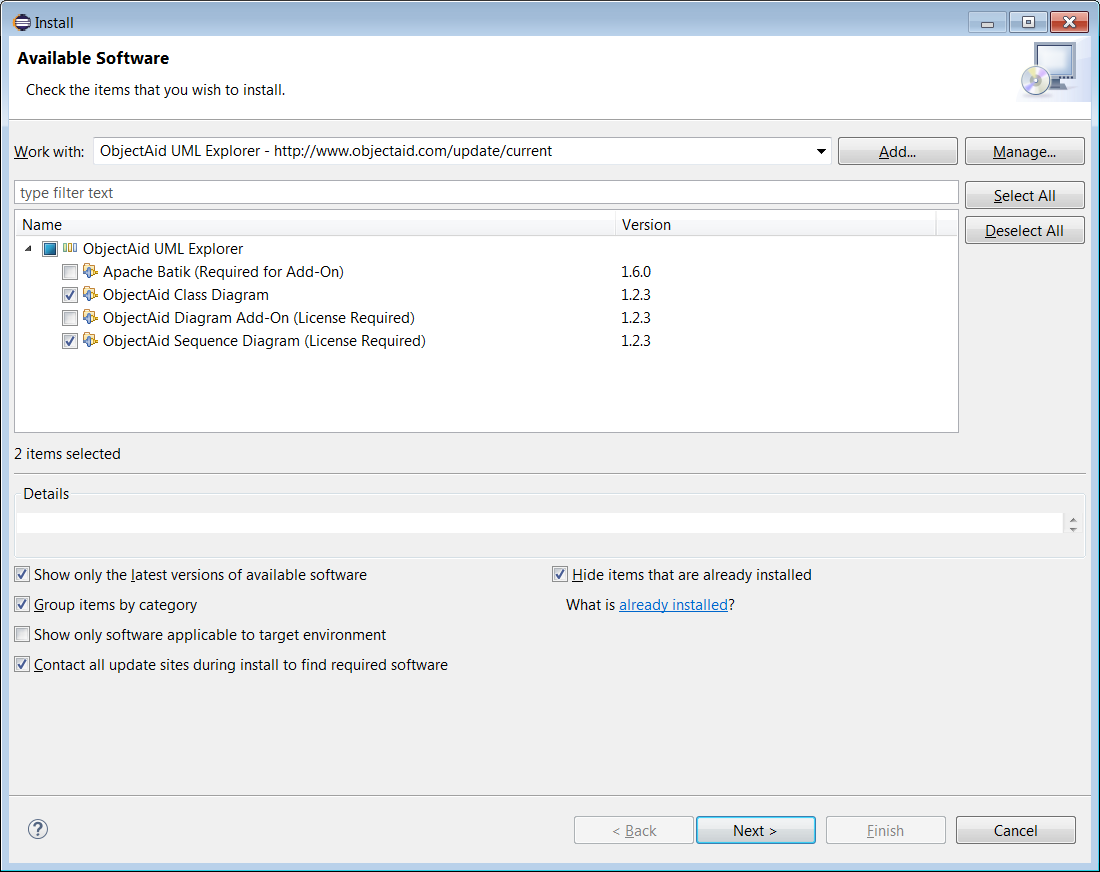
Overview

This homework provides as an introduction to [ObjectAid](http://www.objectaid.com/), an Eclipse plugin to create a class diagrams. ObjectAid does reverse engineering very nicely, and simply; however, it does not do *forward-engineering*.

You will use ObjectAid to create a class diagram.

Download & Install

1. Obtain a license for ObjectAid: <https://www.objectaid.com/obtain-license>.
2. Install ObjectAid in Eclipse: <https://www.objectaid.com/install-objectaid>

As shown on the right, select the Class Diagram and the Sequence Diagram. I’m not sure how the license works now since I already have it installed. However, I can no longer create sequence diagrams as I guess my license expired. Hopefully, you can, as we will have another homework later that you need to create sequence diagrams for.

Directions

1. Unzip the code from the Schedule. The package structure is rather deep:

com\yattasolutions\umllab\examples\shop

All the code is in the *shop* subpackage.

1. Create a project in Eclipse.
2. Drag the *com* folder to the *src* folder in Eclipse. The code will not compile because it is using some proprietary code, I think. However, we can still make a class diagram of it.
3. Select your project folder in the Package Explorer and choose: File, New, Other, ObejctAid UML Diagram, ObjectAid Class Diagram, Next. Give the class diagram a name and choose Finish. The class diagram will be open and empty, nothing on page.
4. Select these classes: *Customer, Employee, Human, Person* and then drag them all at once onto the diagram. Make sure no lines are crossed.

Notes:

* Scroll with the mouse wheel (Not Ctrl+Scroll) and page will zoom.
* If you select a class (highlighted with stronger black border), then this scrolling behavior changes: it now scrolls up and down. I can unselect the class, but scrolling still scrolls up and down.no longer zooms. If you choose a new zoom level from the drop down on the menu bar, the scroll zoom behavior returns.
* When a class is selected you can drag it to a new location.
* You can select an association and move the line, bend it. I couldn’t select a generalization nor implementation arrow.
* Select a class and right-click. There are a number of options. For example: *Show Attributes* and *Show Methods* control which, if any attributes or methods are shown. This can be useful to show multiple versions of a class diagram: one with just classes and relationships, to get an overview of a system; or one with just public methods and no attributes, *etc.*
* If you mess up, select on the diagram, the class you want to remove from the diagram and delete it. You must select the entire class for delete to work. You can result to Ctrl+A to select everything if you want. This only deletes from diagram, not the code itself.

1. Do this on the template below: Describe in complete sentences the relationship between the classes and interface.
2. Drag the *ShoppingCart* class onto the diagram. Make sure no lines are crossed.
3. Do this on the template below: Describe in complete sentences the relationship between the *ShoppingCart* and *Customer* classes.
4. Drag the *Product* class onto the diagram. Make sure no lines are crossed.
5. Do this on the template below: Describe in complete sentences the relationship between the *ShoppingCart* and *Product* classes.
6. Drag the *Customer* class onto the diagram. Make sure no lines are crossed.
7. Do this on the template below: Describe in complete sentences the relationship between the *Company* and *Customer* classes. You do not need to describe the relationship between the *Company* and *Employee* classes.
8. Right-click an empty space on the diagram and choose: Save as image.
9. Save the template below, with your answer to a file named: *hw\_cdoa\_lastName.*
10. Submit the document and image in the HW CDOA dropbox on Blazeview.

**HW\_CDOA Submission Document**

|  |  |
| --- | --- |
| Name |  |

**Save this page as a separate document: *hw\_cdoa\_lastName* making sure to remove the instructions above.**

|  |  |
| --- | --- |
| 6. | Describe in complete sentences the relationship between the classes and interface. |
|  |  |

|  |  |
| --- | --- |
| 8. | Describe in complete sentences the relationship between the *ShoppingCart* and *Customer* classes. |
|  | A customer can have 0 or 1 ShoppingCart and vice versa. |

|  |  |
| --- | --- |
| 10. | Describe in complete sentences the relationship between the *ShoppingCart* and *Product* classes. |
|  |  |

|  |  |
| --- | --- |
| 12. | Describe in complete sentences the relationship between the *Company* and *Customer* classes. You do not need to describe the relationship between the *Company* and *Employee* classes. |
|  |  |