

# 48024 Applications Programming

## Assignment 2

**Topics:** OO Design, GUIs, MVC Pattern, Observer Pattern

**Learning Outcomes:** This assessment task addresses the following subject learning objectives (SLOs): 3, 4 and 5

**Due date:** 11:59PM Sunday the 14<sup>th</sup> of October<sup>1</sup>

**Weight:** 20%

### 1. Individual work

All work is individual. You may discuss ideas, approaches and problems, but you should write every line of code yourself except for code copied from the lecture notes, lecture code or lab code. You MUST NOT let another student see your solution code, and you MUST NOT look at another student's solution code. More information about Academic Misconduct can be found at:

<http://www.gsu.uts.edu.au/rules/student/section-16.html>

### 2. Specification

After the success of Jaime's computer building business your other friend Guillermo is keen to get in on the action and has asked you to build him a similar piece of software to that which you provided for Jaime. Guillermo has higher requirements though, and wants a GUI – no-one types anymore, clicks are sales!

The specification for this GUI is presented in several parts, being an amazing animator, Guillermo has produced a video demonstrating the functioning and appearance of the GUI interface (found on PLATE, with the other assignment 2 material). Guillermo also gives you the following screenshots (also found full size separately on PLATE):

#### Main Menu (20%)



The main menu opens when the program starts. It has an image (provided by Guillermo), a title, two large buttons to open the catalogue and the build respectively, and a small button to quit. The title of the window is also set by the program.

---

<sup>1</sup> Note that the subject outline has the due date as Monday the 15<sup>th</sup>. This is incorrect. Apologies for any confusion in this regard.

### Catalogue Window (30%)

The Catalogue window displays a table of parts with the following data:

Type	Name	Price
STORAGE	evo 860	\$155.00
KEYBOARD	daskeyboard	\$239.00
CPU	i5	\$365.00
MEMORY	corsair 16G	\$299.00
MOTHERBOARD	ASUS ROG	\$159.00
CASE	sheetmetal box	\$39.00
CPU	Ryzen 7	\$299.00

Below the table are filter controls:

Type:

Price Range:  to

Buttons: Add Selected to Build, Add New Part to Catalogue, Remove Selected from Catalogue, Close

The catalogue window shows the parts currently in the catalogue and allows the user to filter this list via three input fields, with accompanying title and labels (see the video for a demonstration of this functionality). It also has several buttons that allow the user to add selected parts to the build, add new parts to the catalogue, remove items from the catalogue and close the catalogue window.

### Add Part to Catalogue Window (10%)

The Add New Part to Catalogue window contains the following fields and buttons:

Type:

Name:

Price:

Button: Add to Catalogue

Allows the entry of details for a new part to be added to the catalogue.

### Error When Adding to Catalogue (5%)

The error dialog box displays the following text:

Invalid price entered.

Button: Okay

Pop up informing user of an error in entering the price for a new part.

### Build Window (30%)

The Current Build window shows a table with the following header:

Type	Name	Price
------	------	-------

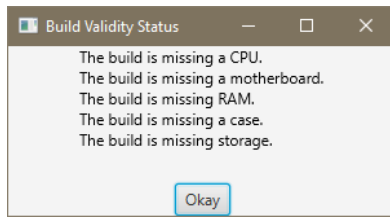
The table is currently empty, displaying the message: "No parts in current build."

At the bottom right, the total cost is displayed: "Total: \$0.00"

Buttons: Check Build, Remove Selected from Build, Close

Shows the user's current build. The total cost of the current build is displayed just below the build, along with three buttons allowing the user to check the build for basic completeness, remove unwanted parts and close the window.

### Check Build Window (5%)



Shows the status of the current build with respect to basic completeness.

## 3. Requirements

### Layout

To achieve full marks, your layout must match the screenshots/video as closely as possible. You do not need to match the OS supplied window frames or widgets, but everything within the window frame is subject to assessment, along with the window title. All spacings are set at 10 in the model solution, with preferred window widths at 500 and 300.

### Style

A CSS file is provided with the skeleton code. This file contains the necessary style components to make your program match the model solution. You may not necessarily need to use all the provided components.

### Code

As a starting point for this assignment, you must use the skeleton code provided on PLATE (<https://plate.it.uts.edu.au/>) under Assessments->Assignment 2. There are two options, a Netbeans version, which includes the structure required to import directly into Netbeans, and a plain version for other IDEs (which tend to be less rigid about their import requirements).

Your solution must satisfy the following code requirements:

- Your solution must employ the MVC architecture.
- Your solution must keep the package and class names provided in the skeleton code.
- The models must notify the views of changes by correctly applying the JavaFX property patterns and observable lists. Model data that can change must be observable. Model data that never changes need not be observable.
- The views must be laid out using FXML.

## 4. Expected workload

The time to do the assignment to a distinction level (i.e. a mark between 75% to 84%) has been estimated at 25 hours for a student of average ability who has completed all the tutorial and lab exercises.

## 5. Online support

The Assignment 2 discussion board has been set up on UTSONline so that students can ask questions, and other students can reply. The course coordinator will only post a reply only if the student response was wrong, or in the case of correcting a mistake in the assignment specification.

**You must not post or share Java code to the discussion board.** The board is there to help you, not to provide the solution. **Posting your code is academic misconduct and will be reported.** Each time this rule is violated, the code will be removed and replaced with a comment of the form: "Strike 1: Posting code". After 3 strikes, the discussion board will be deleted because it did not work.

FAQs (Frequently Asked Questions) and their answers, if enough are asked, will be posted in the Assignment 2 forum. If you have a question, check the FAQ first; it may already be answered there. You should read the FAQ at least once before you hand in your solution, but to be safe check it every couple of days. Anything posted on the FAQ is considered to be part of the assignment specification. The FAQ will be frozen (no new entries) two days before the due date; no questions will be answered after it is frozen.

If anything about the specification is unclear or inconsistent, contact the subject coordinator who will try to make it clearer by replying to you directly and posting the common questions and answers to the FAQ. This is similar to working on the job, where you ask your client if you are unsure what has to be done, but then you write all the code to do the task. Email [luke.mathieson@uts.edu.au](mailto:luke.mathieson@uts.edu.au) to ask for any clarifications or corrections to the assignment.

## 6. Submission to PLATE

### READ THIS ENTIRE SECTION CAREFULLY

Included in the skeleton code is a file called progress.txt which you must fill out as you progress through the assignment. This file will contain lines such as these:

[?] The Main menu window is at least partially done.  
[?] The Main menu window is done.  
[?] The Catalogue menu window is at least partially done.  
...etc...

As you make progress on your assignment, you must edit this file by changing each [?] into a [y] and then submit your progress to PLATE. Don't forget to save this file before submitting. For example, after you get the main menu window partially done (even if you have only done a small amount), you edit this file as follows:

[y] The Main menu window is at least partially done.  
[?] The Main menu window is done.  
[?] The Catalogue menu window is at least partially done.  
...etc...

Then you submit your project to PLATE so that there is a record of what your code looked like when you first started to make progress on your Main menu window. After you complete the Main window feature, you should again update this file as follows:

[y] The Main menu window is at least partially done.  
[y] The Main menu window is done.  
[?] The Catalogue menu window is at least partially done.  
...etc...

Then you submit your project to PLATE again so that there is a record of what your code looked like when you completed this feature.

It is not always required that you complete a feature before moving onto the next feature. For example, your progress.txt file may read:

[y] The Main menu window is at least partially done.  
[?] The Main menu window is done.  
[y] The Catalogue menu window is at least partially done.  
...etc...

This would indicate that you partially completed the Main menu window, then moved on to the Catalogue menu window. This is allowed, as long as you have completed at least enough of the Main menu window that will allow you to correctly open the Catalogue menu window.

### **Important**

If you don't submit your progress on a particular feature, then your marks for that feature won't count! That is, you are only marked for those features where you submit evidence of your progress. Be very careful to always submit your progress as soon as you make progress so that you don't lose any marks unnecessarily.

Your solution is to be submitted to PLATE at <https://plate.it.uts.edu.au/> to Applications Programming/Assessments/Assignment 2. Your assignment should be submitted as a JAR file that includes:

- All Java source files required to compile your assignment.
- All FXML, CSS and image files required to run your assignment.
- The progress.txt file at the top level of your project directory structure.

Based on your submitted progress.txt file, PLATE will calculate a mark. This mark should NOT be considered in any way as your final mark. Rather, it should be considered as a "potential" mark. During the week 12 demonstration and peer marking day, the system will try to assign you to peer mark other students who have a similar potential mark as yourself.

There is no scheduled late submission period. An extension of up to one week may be given by the subject coordinator before the due date; you must supply documentary evidence of your claim. An extension CANNOT be given after the due date.

You may also apply for special consideration for reasons including unexpected health, family or work problems.

More information about how to apply for special consideration can be found at <http://www.sau.uts.edu.au/assessment/consideration.html>.

## **7. Peer marking and demonstration**

In your scheduled week 12 lab class you must be prepared to demonstrate your assignment to your tutor and explain parts of your code to your tutor if requested. If you are unable to explain your code, it may impact your marks. Your presence is required at this class. Any student who is not present without being granted prior permission may have up to 50% of their marks for this assignment deducted.

In addition to demonstrating your assignment, you will also be assigned two other students to peer mark, and two other students will be assigned to peer mark you. The purpose of this peer marking is to mark the functionality of your application which cannot be tested by PLATE. Your marks for

functionality will be based on these peer marks after they are moderated by the subject coordinator. Aside from marks for the functionality, the subject coordinator will also check your code to ensure that all code requirements have been met. If the code requirements have not been met (for example, the observer pattern not used, marks will be deducted for those components). Note that you can only be marked for features that can be demonstrated to work.

Marking the code and analysing spoofing, cheating and plagiarism is done in the two weeks following the due date. If you are suspected of Academic Misconduct, I will forward your case to the Misconduct Committee and will notify you by email. Your mark will be finalised within 2 weeks of the due date.

## 8. Marking scheme

The marks for the assignment are divided into the following functionality components (note that individual tests may test several functionality components, and a functionality component may be tested by several tests):

Functionality Component	Mark Allocation
Main Window	20
<ul style="list-style-type: none"> <li>Image displayed - 2</li> <li>Heading and title correct - 2</li> <li>Buttons correctly laid out and formatted. - 6</li> <li>Buttons open correct windows. - 8</li> <li>Exit button closes entire program. - 2</li> </ul>	
Catalogue Window	30
<ul style="list-style-type: none"> <li>Title correct – 1</li> <li>Headings for table correct – 3</li> <li>Table correctly laid out – 2</li> <li>Filter heading correct – 1</li> <li>Filter labels correct – 3</li> <li>Filter text fields correctly laid out – 3</li> <li>Buttons correctly laid out – 4</li> <li>Buttons function correctly – 5</li> <li>Table updates correctly using observer pattern – 8</li> </ul>	
Add Part Window	10
<ul style="list-style-type: none"> <li>Title correct – 1</li> <li>Labels correct – 3</li> <li>Text fields laid out correctly – 3</li> <li>Button laid out correctly – 1</li> <li>Button functions correctly – 2</li> </ul>	
Error Window	5
<ul style="list-style-type: none"> <li>Title correct – 1</li> <li>Error message correct – 1</li> <li>Button laid out correctly – 1</li> <li>Button functions correctly – 2</li> </ul>	
Build Window	30
<ul style="list-style-type: none"> <li>Title correct – 1</li> <li>Headings for table correct – 3</li> <li>Table correctly laid out – 2</li> </ul>	

---

<ul style="list-style-type: none"> <li>• Price correctly formatted – 1</li> <li>• Total price shown – 1</li> <li>• Total price correctly calculated and updating – 3</li> <li>• Buttons laid out correctly – 3</li> <li>• Buttons function correctly – 8</li> <li>• Table updates correctly using the observer pattern – 8</li> </ul>	
Build Check Window	5
<ul style="list-style-type: none"> <li>• Title correct – 1</li> <li>• Correct text show for current build – 2</li> <li>• Button correctly laid out – 1</li> <li>• Button functions correctly – 1</li> </ul>	

---

This adds to a mark out of 100 and makes up 20% of your final assessment mark.