Part 4 (25 marks): Graphical User Interfaces (GUIs) and Event Driven Programming

This section involves a class called **SearchPortal**. Figure 4.1 shows the default appearance of the graphical user interface (GUI) in the SearchPortal class. When a user clicks on the Clear button, the default GUI appearance is as in Figure 4.1. Figure 4.2 shows the tool text message shown when the cursor is placed over the text field. The SearchPortal accepts a valid course code that must start with a letter and contain 8 characters in total. Figures 4.3 and 4.4 show the changes in the GUI appearance when the Search button is pressed for invalid and valid codes respectively. Refer to the demonstration video on myElearning for more clarification if necessary.

The icons to the right of the textfield in Figures 4.1 (and 4.2), 4.3, and 4.4 are created using a label formatted using the Wingdings 2 font and set to the following characters respectively: B, R, and T. You will need to reference the Java APIs for the textfield and label GUI elements and identify appropriate methods for implementing the functionality (colours, tool tip text) shown in the figures.

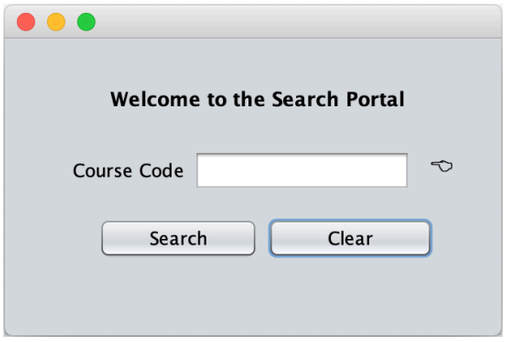

Figure 4.1 Default appearance of the StudentPortal GUI

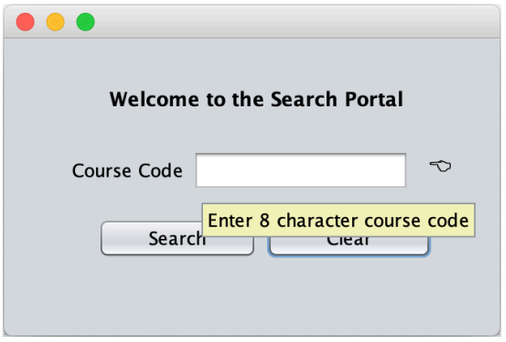

Figure 4.2 Tool Tip Text message when cursor hovers over the text field

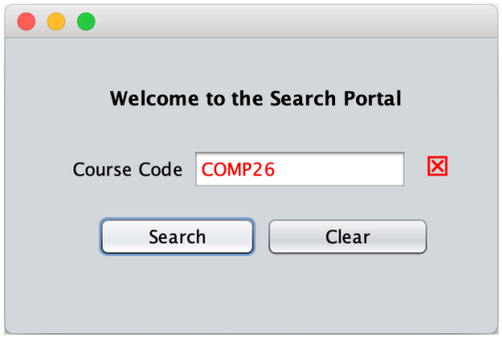

Figure 4.3 GUI appearance when an invalid course code is entered and the Search button is pressed

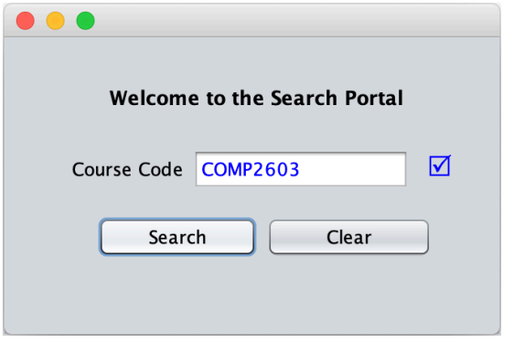

Figure 4.4 GUI appearance when a valid course code is entered and the Search button is pressed

Part 4 continues on the next page..

Part 4 continued..

Define your own custom variable and method names for the various GUI elements in the **SearchPortal** class and:

(a) Write working Java code for a complete method that implements the **Clear** button functionality in Figure 4.1 using event handling code and listeners. [2 marks]

(b) Write working Java code for a complete method that implements the **Search** button functionality shown in Figures 4.3 and 4.4 using event handling code and listeners. [8 marks]

(c) As the user types a course code, any lowercase letters are immediately converted to uppercase and rendered in the textfield. For example, in Figure 4.3, after the user types 'c' but before the user types 'o', the lowercase 'c' is converted to uppercase 'C'.

(i) Clearly identify which subtype of the EventListener interface must be implemented in the SearchPortal class in order to handle the character conversion event described above. Which method in the EventListener subtype must be overridden in order to immediately render the result in the GUI? Explain clearly why this particular subtype must be used instead of the other two options taught in the course. [3 marks]

(ii) Explain how an instance of the EventListener identified in (c)(i) is associated with appropriate GUI component in the Search Portal class. Write working Java code snippets to illustrate your points and explain any obscure logic or IDE (e.g. Netbeans) generated code. [5 marks]

(iii) Write working Java code for the method identified in (c)(ii) that must be implemented to achieve this character conversion feature in the GUI. Explain clearly why this particular method is used over other options (if any). [4 marks]

(d) Explain clearly, using valid code snippets, how the tool tip text hover functionality in Figure 4.2 is achieved. Reference all of the appropriate EventListener and event handling methods involved in the SearchPortal class your answer. [3 marks]

End of Part 4
Please turn to the next page