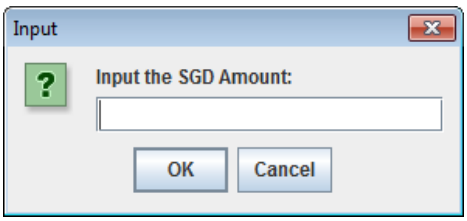
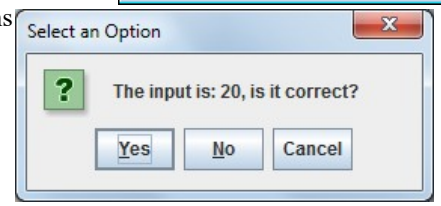
Question 1

(a) Write a class `DialogBox` containing the following methods:

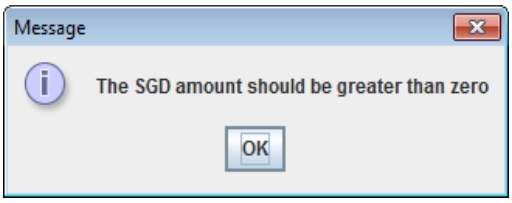


(i) `inputSGDAmount()` : which displays a dialog box asking user to "Input the SGD Amount: " and finally returns the input as a **real number**.

Before the returning the input, ask the user to confirm using a dialog box, which contains "The input is: <user input>, is it correct?", where "<user input>" is the input of the user. If the user does not confirm "yes", return zero. Copy the **class**, including import statement(s), as the answers to this part.



(ii) `checkSGDAmount()` : which calls `inputSGDAmount()` to get the SGD amount to check and nothing is returned. The SGD amount should be greater than zero. If not so, display the error message "The SGD amount should be greater than zero" and then call `inputSGDAmount()` to get the SGD amount again until it is correct. You need to use a loop to achieve this. Copy the **method** as the answers to this part.

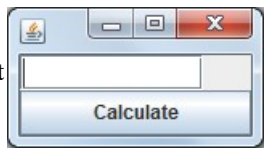


(iii) `main()` : which creates a `DialogBox` object and calls the method `checkSGDAmount()` for testing. Copy the **method** as the answers to this part.

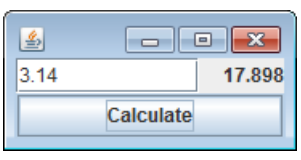
[10]

(b) Write a class `Conversion` containing the following methods:

(i) constructor : which builds the frame shown on the right side. The frame consists of a text field for inputting a SGD amount, a label with 10 spaces for an equivalent SGD amount in HKD, and a button to start the calculation. Declare any necessary attributes in the class and add appropriate action listeners for future use. Copy the **class**, including import statement(s), as the answers to this part.



(ii) `actionPerformed()` : which performs the calculation and puts the result on the label when the button is pressed. You can assume one SGD is equivalent to 5.7 HKD. You can assume a valid real number is entered in the textfield. Copy the **method** as the answers to this part.



(iii) `main()` : which creates a `Conversion` object and sets it visible for testing. Copy the **method** as the answers to this part.

[20]

Question 2

(a) Create a class `Router` which stores the information of a router. It includes the brand, the model number (`String`) and the price (`double`, in dollars). Write a constructor of the class to so that the information mentioned is initialized when a `Router` object is created. Also write the getter methods for those variables. Finally add a method `toString()` to return the router information in the following string form.

```
"brand: Linksys, model number: RVS4000, price: 1080.0"
```

Copy the content of the **class** as the answers to this part.

[6]

(b) Create a class `ComputerShop` which stores the router information in a map `routerMap`, whose key is the concatenation of the brand and model number, separated by `": "` (a colon and a space). The values of the map are the `Router` objects. Write a method `addRouter(Router oneRouter)` which adds `oneRouter` to `routerMap`. Copy the content of the **class**, which any include import statement(s) required, as the answers to this part.

[3]

(c) Create a class `TestComputerShop` with a `main()` method which creates a `ComputerShop` object `aShop` and add the first router with brand "`Linksys`", model number "`RVS4000`" and price `1080`. Add the second router with brand "`Planet`", model number "`VRT-311S`" and price `510`. Copy the content of the **class** as the answers to this part.

[4]

(d) Write a method `showRouter()` of `ComputerShop` which loops through the keys of `routerMap` using the enhanced for-loop and directly prints each router object stored using `System.out.println()`. (Loop through the values is simpler but using the keys is required in this part.) This should show suitable information since the method `toString()` has been written in (a). Add a statement in `TestComputerShop` to display all the router information of `aShop`. Copy the content of the **method**, **line(s) added** and execution output as the answers to this part.

[4]

(e) Write a method `modelNumberSet()` of `ComputerShop` which returns model numbers of the routers in a set. You should loop through the values of `routerMap` using the enhanced for-loop and collect the model numbers. Add a statement in `TestComputerShop` to display the set using `System.out.println()`. Copy the content of the **method**, **line(s) added** and new execution output as the answers to this part.

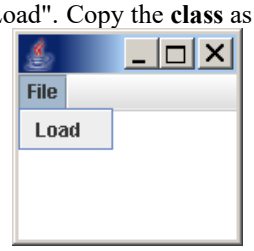
[6]

(f) Write a method `priceList()` of `ComputerShop` which returns the prices of the routers in a list. You should loop through the values of `routerMap` using the enhanced for-loop and collect the prices of the routers. Add a statement in `TestComputerShop` to display the list using `System.out.println()`. Copy the content of the **method**, **line(s) added** and new execution output as the answers to this part.

[2]

Question 3

(a) Create a class `HexEditor` with a constructor which creates a 5 x10 text area in a JFrame. Also add a pull-down menu with a menu item "Load". Copy the **class** as the answer to this part.



[6]

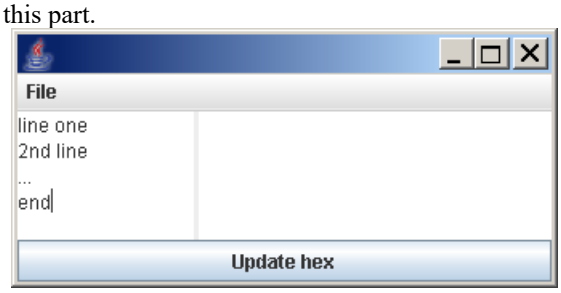
(b) Create another class `TestHexEditor` with a `main()` method which creates an object `anEditor` of the class `HexEditor` and displays the frame in part (a) using `setVisible(true)`. Copy the **class** as the answer to this part.

[2]

(c) Make changes to the class `HexEditor` so that it implements `ActionListener`. In the constructor of the class `HexEditor`, add the current object as the action listener of the "Load" menu item. Write the corresponding method `actionPerformed()` which displays a dialog box asking for a file name and load content of the file to the text area using `FileInputStream`. You can assume the file is a byte file and it exists. Copy the **changed/added lines and the new method** as the answers to this part.

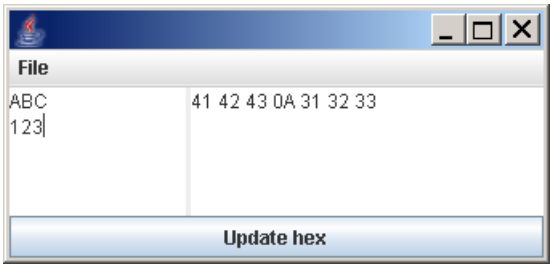
[8]

(d) We will use the BorderLayout manager. Modify the constructor to put the text area on the left, add a label containing a space in the middle, add a new text area (with width 20) on the right, and a button at the bottom. The resulting window is showed below (after a file is loaded). Copy the **changed/added lines** as the answers to this part.



[3]

(e) Modify the method `actionPerformed()` so that when the update button is pressed, a hexidecimal version of the file is displayed on the right, in which each byte is represented by two hexidecimal digits and followed by a space. You can use `Integer.toHexString('B')` to convert a byte `'B'` to a hexidecimal string. A sample window is shown below (where `'0A'` on the hex window corresponds to a newline character). Copy the **changed/added lines** as the answers to this part.



[5]

(f) In the pull-down menu, add a "Save" menu item so that the original file can be replaced by the content of the text area on the left. You need to modify the method `actionPerformed()` to achieve this. If no file was loaded but the user directly type something on the left text area, ask for the file name using a dialog box and replace any existing file. Copy the **changed/added lines** as the answers to this part.

[6]

Question 4

(a) Using one SQL statement to perform each the following:

(i) create a table "item" storing item number (10 characters, primary key) and price (8 digits with 2 decimal places)

(ii) insert the data ("Milk-01", 13.8) into the table

(iii) display the item information with price not less than 10.

[6]

(b) Write a class `Database2Txt` with a method `database2Txt(String filename)` to retrieve data from the database table "item" in part (a) and store the information in a text file with name `filename`. Each row of data from the table should be stored in one row of the text file, with data separated by a space.

As far as possible, use the attributes and methods (e.g., `Connection conn`, `loadDriver()`) taught in our lectures.

[9]

**** End ****