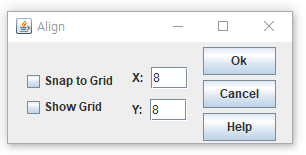
**OOP Lab 13**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: |  | Department: |  |
| Student ID: |  | Room Number: |  |
| Due Date: | **June 19, 23 : 59** | | |

**Submit your assignment using the following file format:**  LabNumber\_StudentName.zip (eg. Lab12\_Hongkildong.zip). This zip file will contain **two types of** files, namely: **report file** with file format **“Report\_Lab number**” (eg. report\_12) to answer theory questions and to write the screen shot of your program and Source code file that contains codes of classes to answer programming questions.

**Q1.** The Skeleton code to create the following GUI without any functionality is given. Hence, complete the skeleton code in files “Align.java” and “AlignFrame.java” under the folder CodeQ1. After completing the code, when you run the program, the following window is displayed.



**Requirement 1:** The title of the frame is Align

**Requirement 2:** The size of the frame is 300x140

**Hint:** Steps to put the above GUI components in a frame

Step 1: **Creating two Check boxes**

1. Create the two Check boxes:”Snap to Grid" and “Show Grid".
2. Put them inside “**panel\_1”** which uses GridLayout( 2,1) as its layout manager.

Step 2.1: creating Jlabel **and JTextField**

1. Create Text Field with 8 and its label X ..
2. Put them inside “**panel\_2”** which uses “**FlowLayout”** as its layout manager

Step 2.2: creating Jlabel **and JTextFiled**

1. Create Text Field with 8 and its Jlabel Y .
2. Put them inside “**panel\_3”** which uses “**FlowLayout”** as its manager

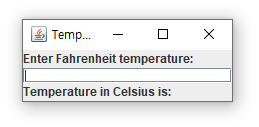
Step 2.3: Put p**anel\_2** and **Panel\_3** inside **Pane\_4** which uses **Borderlayout()** as its layout manager**.** So, put **panel\_2** inside **panel\_4** at North and **panel\_3** inside **panel\_4 at** South.

**Step 3: Creating three Buttons**

1. Create “**Ok”** button.
2. Create “**Cancel”** button
3. Create “**Help” button**
4. Put them inside “**pane1\_5”** which uses “**GridLayout(3,1,10,5)** as its layout manager

**Step 4: Put Panel\_1, panel\_4 and panel\_5** inside the frame using **FlowLayout**(FlowLayout.CENTER, 10, 5));

**Q2**. The Skeleton code of an application that converts from **Fahrenheit** to Celsius is given. The Fahrenheit temperature should be entered from the keyboard via a **JTextField**. A J**Label** should be used to display the converted temperature as shown in the following figure. Use the following formula for the conversion: **Celsius = (5/9)\*(Fahrenheit – 32). Complete the skeleton codes in files “Convert.java” and “ConverFrame.java” under “Code Q2” folder**. After completing the code, you will get the following window when you run it.



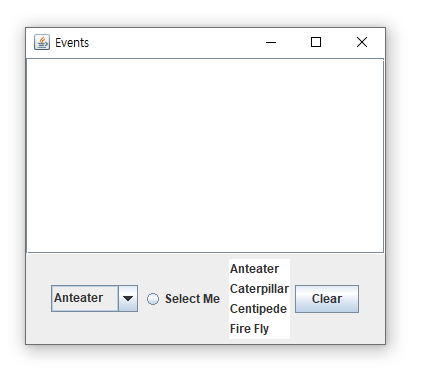
* **Requirement 1:** The title of the frameTemperature converter
* **Requirement 2**:The size of the frame is 225 x 90
* **Requirement 3:** The frame uses BorderLayout manager to put the following three components
* **Requirement 4**: Put JLabel with “**Enter Fahrenheit temperature**” at north location
* **Requirement 5**: Put JTextField( blank space) at Ceneter location and its length is 10
* **Requirement 6** : Put JLale with “Temperature in Celsius is: “ at south location
* **Requirement 7:** After entering value of **Fahrenheit and press “enter key” .**Then the output will be displayed on the above window

**Q3**. **Program with one Event Handler class that implements many types of Listener Interfaces.**

* The program display the events that occur during the execution of an application.
* This can help you to understand when the events occur and how they’re generated.
* The source code of an application that enables the user to generate and process every event discussed in this chapter is given in files “Events.java” and EventsFrame.java.
* The application provides methods from the following interfaces to display messages when the events occur. **toString()** method is used to convert the event objects received in each event handler into **Strings** that can be displayed. **toString (**) method creates a String containing all the information in the **event** object.

1. **ActionListener** Interfcae
2. ItemListener Interface
3. ListSelectionListener Interface
4. MouseListener Interface
5. MouseMotionListener Interface
6. KeyListener interfaces
7. **Complete the partial code in the file in** “Events.java” under “CodeQ3” folder.

* **Requirement 1**: The title of the frame is Events
* **Requirement 2**: The size of the frame is 375 X325
* **Requirement 3**: After completing the code in the file “**Events.java**”, when you run the program the following figure will be displayed.



1. After **answering question a), answer the following questions.**
2. Select from **Combox** box and then add the screen shot in your file.
3. Click the “**clear”** button and then add the screen shot in your file.
4. Click the ”**select me**” radio button and then add the **screen shot** in your file.
5. Select from “**list**” and then add the screen shot in your file.

* **Note :** Since you have no time, the source code in the “EventsFrame.java” is given to you instead of asking you to write it. This source code helps you to under chapter 11 and chapter 12 .

Good Luck for Your Final Exam