

Exercise 1:

[10 points] Write a program that shows a frame with a button labeled “Date and Time” and a text field. Whenever the user clicks the button, the current date and time should be displayed in the text field. You can use:

```
GregorianCalendar dateTime = new GregorianCalendar();
```

to get an object containing current date and time.

Exercise 2:

[10 points] Write a program that shows a frame with three buttons labeled “Red”, “Green”, and “Blue” and a label containing an icon with a circle that is initially red. As the user clicks the buttons, the fill color of the circle should change. When you change the color, you need to invoke the repaint method on the label.

Exercise 3:

For this exercise, you can re-use code examples presented in class, as needed.

- a. **[30 points]** Write a class `ClockIcon` that implements the `Icon` interface type. Draw an analog clock with hour, minute, and second hands showing the current time. To get the current time, construct an object of type `GregorianCalendar` with the default constructor. Look up documentation of the `Graphics` class to see how to draw a line. (Note, to complete this part of the assignment, you may have to refresh your memory for how to determine x and y coordinates of a point on a circle.)
Write a `ClockIconTest` class with a `main()` method to test your `ClockIcon` class .
- b. **[20 points]** Add `javax.swing.Timer` object to your program. The timer’s action listener should invoke the `repaint` method on the label containing `ClockIcon` once per second.
- c. **[10 points]** Modify your program to have three instances of `ClockIcon` (with a timer that invokes `repaint` method once per second). The three clocks should show current time in three different time zones: local, GMT, and Tokyo.

Exercise 4:

[20 points] Look up and read about `Comparable` interface type. Create a `BankAccount` class that implements `Comparable` interface. Order bank accounts by increasing balance. Supply a test program that sorts an array list of bank accounts. You can use `java.util.Arrays` class to do the sorting as follows:

```
BankAccount [] accounts = new BankAccount[numBankAccounts];

for (int i = 0; i < numBankAccounts; i++){
    accounts[i] = new BankAccount (...);
}

Arrays.sort(accounts);
```