

VERY COOL

The JOptionPane class a useful class consisting of mostly static methods. For this unit you'll work with simpler versions of the showConfirmDialog, showInputDialog, and showMessageDialog methods. Notice that these methods are overloaded. Here is a short sample program. Copy it and run it.

```
import javax.swing.JOptionPane;

public class RunOptions{
    public static void main( String [] args ) {
        JOptionPane.showMessageDialog( null,
            "The parent component will always be null in this unit." );
        JOptionPane.showMessageDialog(null, "The return type is void", "Overloaded",
            JOptionPane.WARNING_MESSAGE);

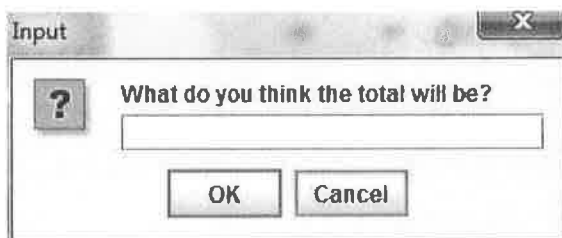
        String str = JOptionPane.showInputDialog(null, "What's your name?", "Goofy" );
        JOptionPane.showMessageDialog( null, "Hi " + str );

        int response = JOptionPane.showConfirmDialog( null, "Is that your real name?",
            "Confirm", JOptionPane.YES_NO_CANCEL_OPTION,
            JOptionPane.QUESTION_MESSAGE );

        if ( response == JOptionPane.YES_OPTION )
            JOptionPane.showMessageDialog( null, "You said YES");
        else if ( response == JOptionPane.NO_OPTION )
            JOptionPane.showMessageDialog( null, "You said NO");
        else if ( response == JOptionPane.CANCEL_OPTION )
            JOptionPane.showMessageDialog( null, "You hit CANCEL");
        else
            JOptionPane.showMessageDialog( null, "You hit X. Very suspicious");
    }
}
```

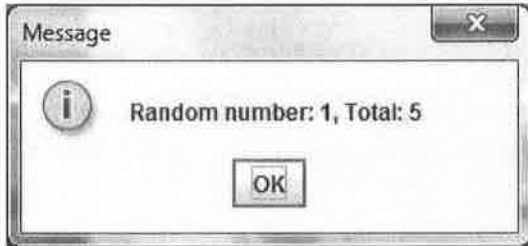
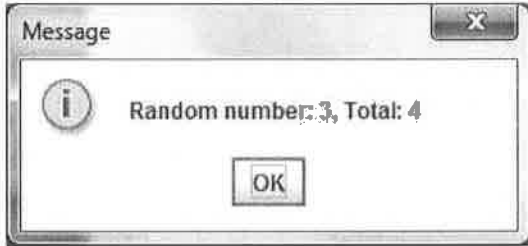
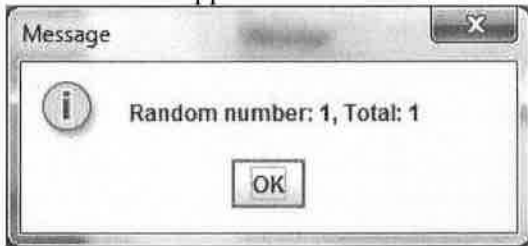
Check out the api at <http://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html> for some more information (though there is a lot more there than you will need).

2. Write a program where the program will generate three integers (with random values between 1 and 5). The user guesses what the total will be. You will need to use the showMessageDialog and showInputDialog methods. It should look something like this.

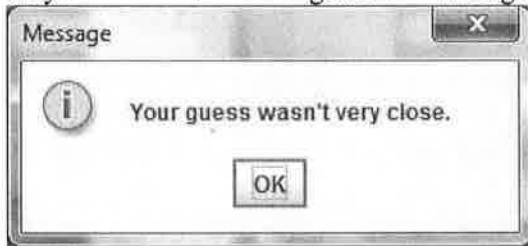


In this test case I entered "9". If the user clicks Cancel or the Close button then a message dialog should appear saying something like "Game Over." When that dialog is closed the program is over.

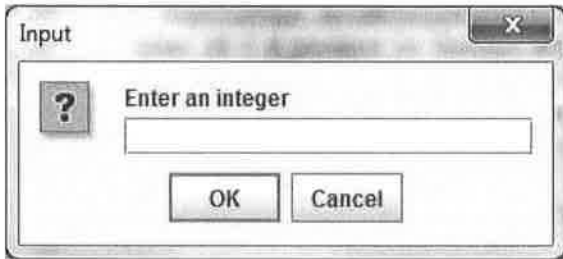
In this example the three random numbers were 1, 3, and 1. Message dialogs like the ones shown should appear.



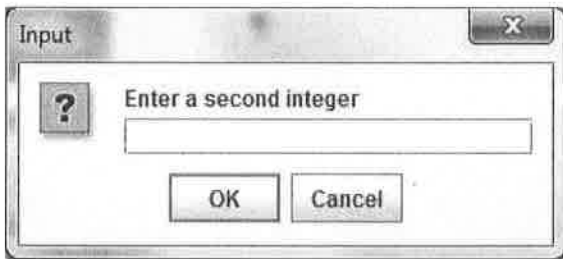
If the user guessed the correct total exactly, then the final message should congratulate them. If their guess was more than 2 units off (above or below) then display a message indicating that they were close. If their guess was wrong by 3 or more (as in this example)



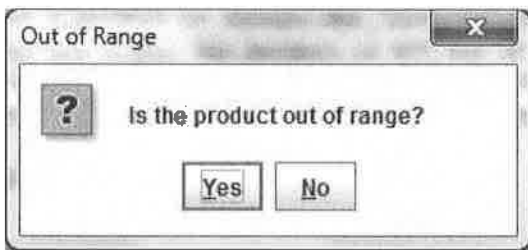
3. The user enters two integer values using the JOptionPane class. The program then asks the user if the product will be out of range. Remember, an int can only have a value between roughly +/- 2 billion. Your program should start like this:



Assume the user enters an integer and clicks OK.



Assume the user enters an integer and clicks OK.



Your program will then display one of five possible responses.

- Correct because the user clicked YES and the product is out of range.
- Correct because the user clicked NO and the product is not out of range.
- Wrong because the user clicked YES and the product is NOT out of range.
- Wrong because the user clicked NO and the product is out of range.
- Chicken because the user closed the dialog by clicking on the X.

Here is half-a-hint. Enter the following code after the section where the user enters two numbers.

```
double d_product = n1*n2;  
int i_product = n1*n2;  
JOptionPane.showMessageDialog( null, "double: " + d_product + ", int " + i_product );
```

I say half-a-hint because the above code has a flaw. `d_product` and `i_product` always have the same values. Find and fix the flaw and the rest of the program should be straight forward. Or ignore this and find your own approach.