**Institute of Technology Tralee**

**Computing Department**

**Object Oriented Programming 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Tutorial 8 – Java Loops and Input Validation**

1. A Java program is required which will ask the user to enter the amount they wish to invest in a bank (their initial balance), the rate of interest available and the number of years they wish to invest for. You can take it that the number of years for investment will be a whole number, while the other quantities can be fractional values.

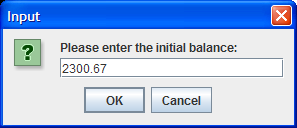
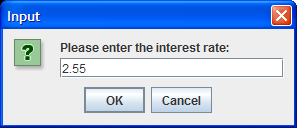
Once this information is provided, the program should then display the balance at the end of each year to 2 decimal places, using a **for** loop. A **JTextArea** component should be used here to display the output and its initial size should be set to 15 pixels high and 40 pixels wide. The font for the text-area should be set to a “monospaced” type, a plain font style and a point size of 12.

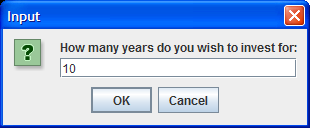
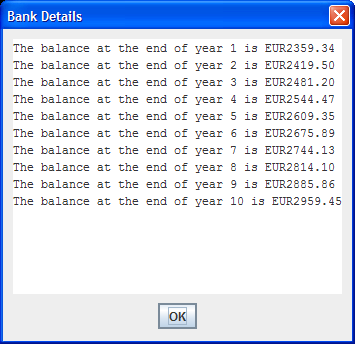
To calculate the balance at the end of each year, the following formulae can be used:

**interest = balance x (interest rate ÷ 100)**

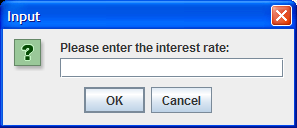
**balance = balance + interest**

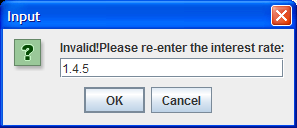
Your program would run as indicated in the following sample screenshots:

1. The program you wrote in part(a) should now be modified so that, rather than the initial balance getting read in to the program in main(), it is done instead via a user-defined method called **getInitialBalance**(). This method should take no arguments but should return, as a floating-point value, the initial balance entered by the user on the dialog. It should be called within main() at the appropriate location, so that the program runs exactly the same as before.
2. The program as it currently stands lacks any input validation and so could easily crash at runtime if invalid data is supplied. You should now modify the program so that the **interest rate** will be validated to a large extent. Recall that the interest rate can be a positive fractional value here. You don’t have to validate against the user entering just “.” here and you can also take it that the user will only ever enter values within the allowable limits for Java float variables for the interest rate. The following sample screenshots show what should happen when the user supplies invalid interest rate values:

**…….Eventually the user supplies a valid interest rate (2.89) and then the program moves on to ask them for the amount they wish to invest**

