**Institute of Technology Tralee**

**Computing Department**

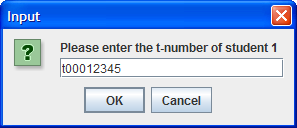
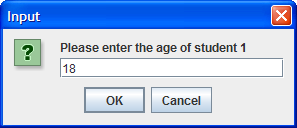
**Object Oriented Programming 1**

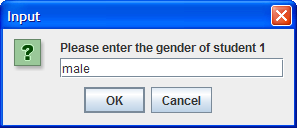
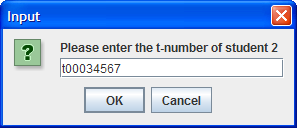
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Solving 4 – Loops and Input Validation**

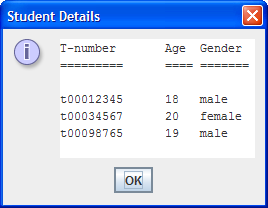
**Q1.**

1. A Java program must be written that processes the details of exactly 3 ITT students (use a **for** loop for this process). The program needs to read in each students’ *t-number*, *age* and *gender*. Once the details have been entered, you should display them in a nice, neat fashion via a text-area, as indicated in the sample screenshots below. You can take it that the font used by the text area is “monospaced” and that the font style is plain with a point size of 12.

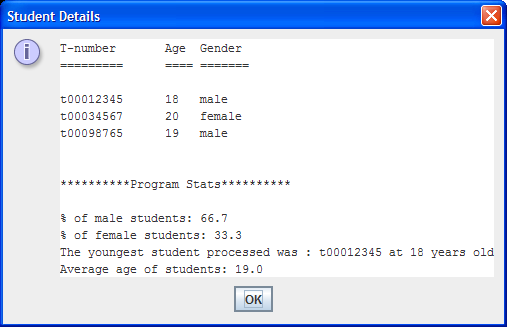
**….. the other 5 pieces of input follow and then the message dialog appears as follows:**



1. Your program should now add to the functionality of part (a) by adding appropriate code so that the program will also display the following results after the loop has completed:

* percentage of male students processed to 1 decimal place
* percentage of female students processed to 1 decimal place
* t-number and age of the youngest student processed
* average age of the students processed to 1 decimal place

Given the same inputs as in part (a), your program will now run as follows:

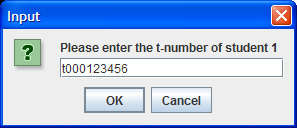
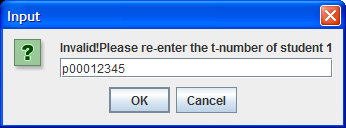


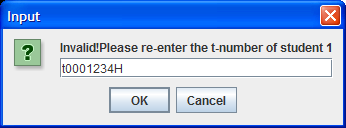
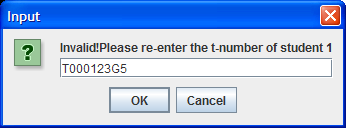
1. Your program should now modify the code from part (b) so that the determination of the average age of the students should be carried out through a **user-defined method** called averageAge(). This method should take a single integer argument which will store the total age of all the students and the method should return the average age as a float.

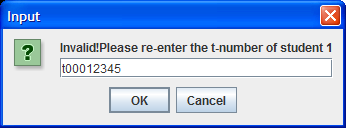
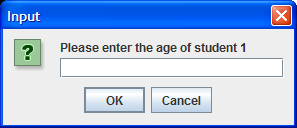
The program would run exactly the same as for part (b) given the same inputs.

1. Your program should now build on part (c) so that the t-number gets validated properly. In this case a valid t-number will have exactly 9 characters, the first one will begin with the letter ‘t’ (lowercase **or** uppercase) and then be followed by exactly 8 digits. Should an invalid t-number be entered, the program should issue the user a warning message and keep issuing this message until a valid t-number has been entered.

The screenshots below show what will happen when the user enters some invalid t-numbers. In this case 4 bad t-numbers are entered before a valid one is given:

1. The program should now build on part (d) to ensure that only valid gender values are accepted. In this case we will say that the only valid gender strings are “male” and “female”. Anything else should be rejected, with the user getting an appropriate warning message until a proper value is supplied.

The screenshots below show what will happen if invalid gender values are supplied:

