3815ICT-Software Engineering Workshop 7

**Activity 1**

Study the entry in Wikipedia for the “Inversion of control” (en.wikipedia.org/wiki/Inversion\_of\_control). Find two aspects of this Wikipedia page that are similar to the material for this topic in the lecture notes.

**Response 1**

Both the Wikipedia page and the lecture slides credit Inversion control with an increase in modularity within the given program.

They also provide information about the way inversion control provides decoupling between tasks and implementation and the goal of creating modules which focus solely on the task it is designed for.

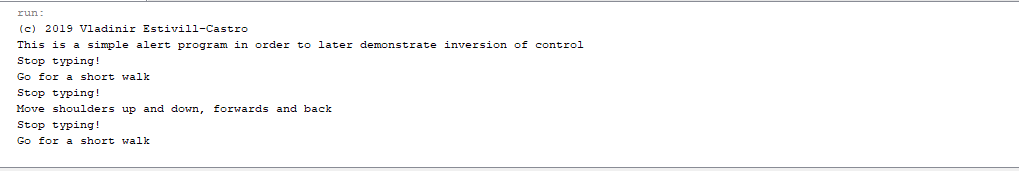
**Activity 2**

In the lecture slides we presented three (3) versions of a simple application for a typing break reminder program.

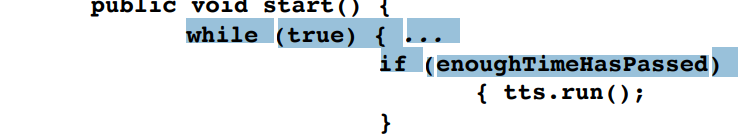
Figure 1 is the class diagram of first version as per the lecture slides.

Review the lecture slides for this version and find the code snippets provided in the lecture slides. Then download the Netbeans project titled SimpleAlert. Run the code of this project, obtain a screen shot of the console where every 4 seconds the program suggests a new exercise. What are the differences between the code in the lecture slides and the code provided in this Netbeans project

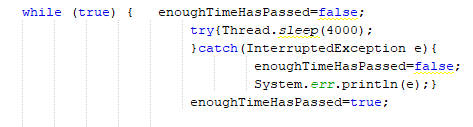
**Response 2**



There are quite a few differences between the code provided in the workshop and the example in the lecture. For example, the SimplerTimer file uses a different approach. One of them runs with a statement



This is different in the program provided to us, the difference is superficial but sets the program up to be written differently.



Also, there is a variable which is set to random which might be used to randomise the order of the output when the program is running.

**Activity 3**

In the second version of the design, an abstract class is introduced. In the code here, we named *TimedTask*, to reflect that the objects of subclasses of this abstract class are tasks that are timed. This is slightly different than in the lectures. What is the name used in the lectures? Argue in favour or against the choice of name in the lectures or the choice used here.

Figure 2 is the class diagram of the second version as per the lecture slides.

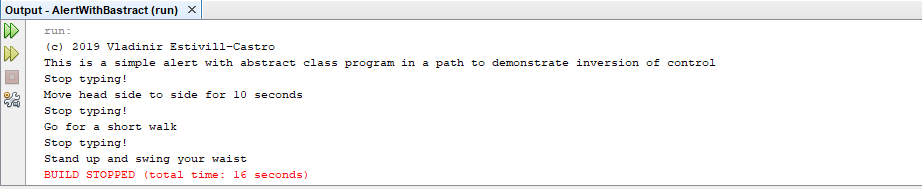
Review the lecture slides for this second version and find the code snippets provided in the lecture slides. Then download the Netbeans project titled *AlertWithBastract*. Run the code of this project, obtain a screen shot of the console where every 4 seconds the program suggests a new exercise. What are the differences between the code in the lecture slides and the code provided in this Netbeans project.

Build a new project that corrects the name of the project from *AlertWithBastract* to *AlertWithAbstract*. Preferably build the project from scratch, and change the suggested exercises to something you think would be useful to do to break being typing for to long on a computer.

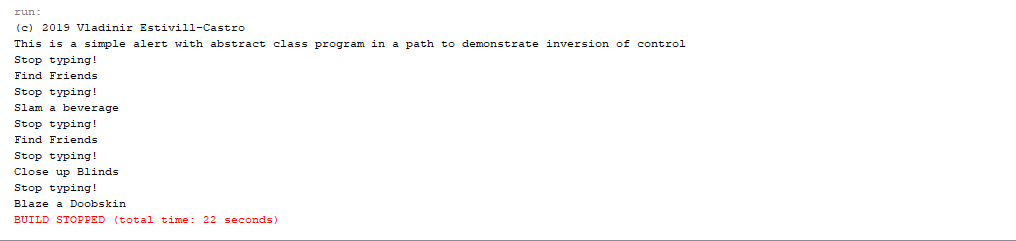
**Response 3**

The lecture notes have this class as TimerTask

It makes sense to call this class TimedTask because each task in being timed and this is a clear way to identify the functionality of the class.



I set up a new project named *AlertWithAbstract and added custom messages. The output is:*



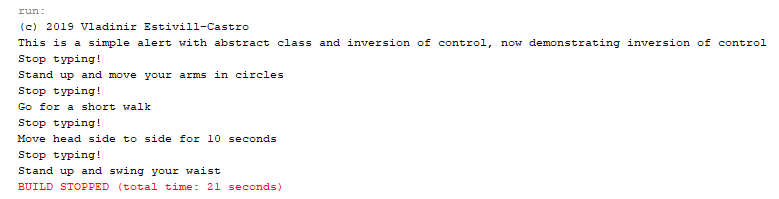
**Activity 4**

In the third version, the pattern of Inversion of Control is presented. Figure 3 is the class diagram of the third version as per the lecture slides.

Review the lecture slides for this third version and find the code snippets provided in the lecture slides. Then download the Netbeans project titled *AlertWithInversionOfControl*. Run the code of this project, obtain a screen shot of the console where every 4 seconds the program suggests a new exercise. Again, discuss the differences between the code in the lecture slides and the code provided in this Netbeans project.

Comment of the three versions so far and mention at least one advantage and one disadvantage of the design. Make sure to discuss issues such as coupling and cohesion.

**Response 4**



There are no obvious differences between the code provided and the code used in the lecture slides. The three versions of this code all accomplish the same task, however, some of them do it more efficiently than others. The most recent iteration may be slightly bulkier but has the advantage in the way it has been set up. Using inversion control has separated the responsibilities of the tasks and the code is now much more reusable. If this program was to be expanded upon, it would be much easier in the third iteration to do so. The simple task on the other hand is simple to write but expanding upon it would require a potential overhaul.

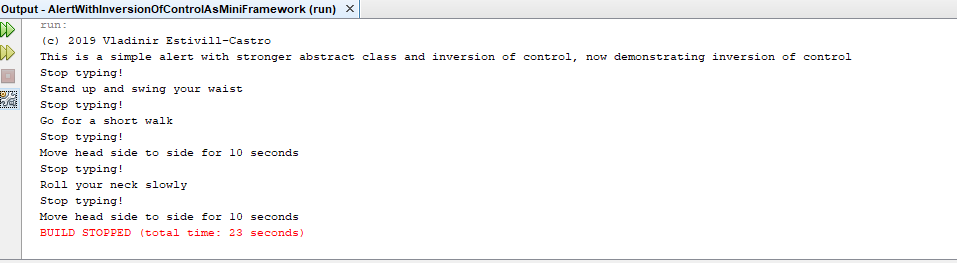
**Activity 5**

Download the code in the Netbeans project titled *AlertWithInversionOfControlAsMiniFramework*. These code goes beyond the three versions in the lecture slides. Run the code of this fourth version of project, and also obtain a screen shot of the console where every 4 seconds the program suggests a new exercise.

What is new in this project? Do you agree with the new design? What are the disadvantages and what are the advantages.

Use the *easyUML* facility of Netbeans to obtain class diagrams for the four versions. The corresponding class diagrams using this tools are shown in Figure ??. You should obtain class diagrams like these for the 4 version of the project; however, for the first 3 version you must draw them to correspond in their layout to the slides in the lecture notes. Include in your submission these version of the class diagrams for the 3 versions. Use the diagrams to emphasize further your arguments about coupling and cohesion from the previous activity.

**Response 5**



I am unable to obtain class diagrams as easyUML is not accessible on the school computers.

**Activity 6**

Draw the collaboration diagram of version 3 or version 4 of the projects discussed in this workshop/lab.

**Response 6**