General instructions:

- In the exam, you will be allowed to copy files to your exam machine during the first 15 minutes of the examination period.
- No documents apart from ones stored in the exam machine can be accessed at any time during the examination period.
- Internet access is not allowed at any time during the examination period.
- Use **c:\temp** as Eclipse workspace for all projects in the exam.

For this problem:

- Expected completion time: 30 minutes

Practice problem 3: LetterShooter

1. **Java project creation**: (For the real exam)

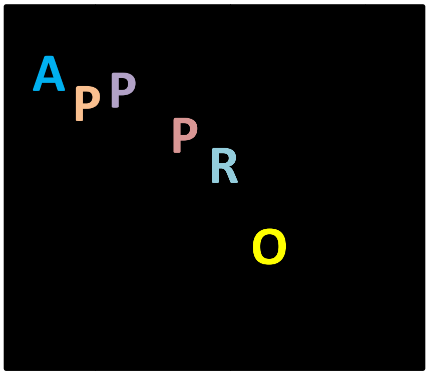
   - Failure to finish this part will prohibit further grading of this problem.

   a. Set up a new Java project in Eclipse named "practice_problem3" so that the location of the project is at **c:\temp\practice_problem3** (We will only grade this problem by opening the project from this folder.)

   b. You are provided with **LetterShooter.java**.

2. **Program description**: The *LetterShooter* Java application is a *JFrame* application that lets the user shoot letters from the bottom of the application frame by pressing keys on the keyboard.

   - The window size is 800px X 800px, non-resizeable.
   - When a key is "released" (activating *keyReleased(KeyEvent)* of the corresponding listener), the letter corresponding to the key (such as 'A') will be seen flying from the bottom of the frame in an upward direction as if it was shot from a shooter.
   - The following picture shows an example of the screenshot of the program when letters in "APP PRO" were typed quickly. In the picture, the letters are moving upward (probably with different speeds).



3. **Write the code**:

   a. In completing the program, adhere to these items:

   - The given *ShotLetter* class must be used to represent the letter being shot.
   - An instance of the *ShotLetter* class must represent a single letter.
   - A separate *Thread* object must be used to update the *y* position of each *ShotLetter* instance. We recommend that you utilize the ***moveThread*** attribute of the *ShotLetter* instance to perform this job. If so, each letter can take care of painting itself on to the *Graphics* object of the *DrawingPanel* instance (via *draw(Graphics)* of *ShotLetter*).
   - Each letter is moved upward by the number of pixels equals to the value of its ***speed*** attribute every ***MOVE_RATE*** milliseconds.

   b. Modify ***startMove()***, ***stopMove()***, and ***run()*** so that the program works as described above.

   - ONLY the 3 methods can be modified.
   - Neither additional classes, attributes nor methods can be added to the source code.

Practice problem 3: Grading Checklist

Note that this is NEITHER a set of grading criteria NOR an answer sheet. Don’t write anything on this page.

The project can be opened properly.	
The program can be compiled and executed.	
The letters moves accordingly when the keys are released.	
The program works without throwing exceptions.	
The modifications made to the source code are limited only to the allowed methods.	