CSCI1530 Computer Principles and Java Programming 2014-2015 Second Term
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Due date: 27 February 2015 (Fri) **Assignment 3** Full mark: 100
Expected normal time spent: 5 hours

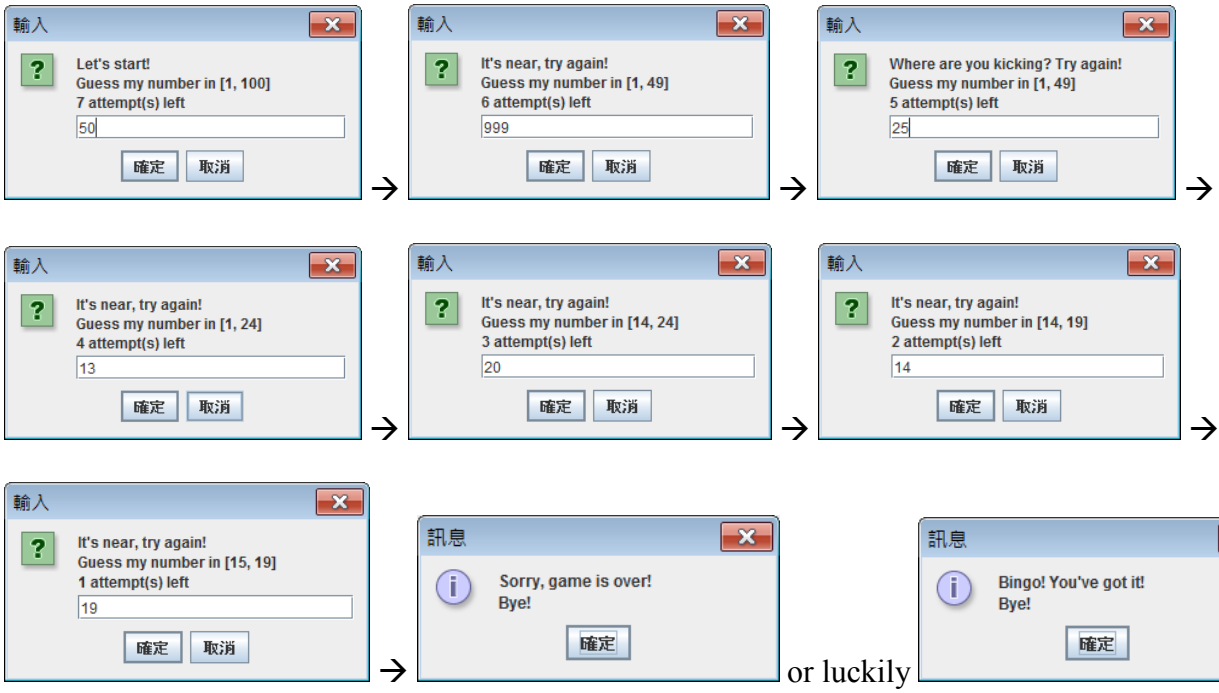
Number Guessing Game (射龍門 – 估數字遊戲)

Aims: 1. Implementing an interesting game.
2. Practising flow control with if-else branching and while looping.
3. Random number generation.
4. Dialog with the user and HTML String output.

Requirements:

1. We are going to implement a simple number guessing game using Java. The program randomly generates a target integer and lets the user guess the number within the range [1, 100]. The user may have at most 7 attempts. On each unsuccessful attempt, hint will be given. If an unsuccessful guess falls within the range, unlikely numbers will be rejected to shrink the range.

2. The program interacts with the user using dialog boxes:



3. The program pops up input dialogs using system class **`JOptionPane`**. Such dialog supports displaying text with HTML tags. It captures user input in a returned String. We may then convert the user input String into an integer using **`Integer.parseInt()`**. Usage example:

```
String input;
// pops up an input dialog in HTML; the String starts with a <html> tag
input = JOptionPane.showInputDialog( "<html>Line1<p>Line2<p>" );

int number;
// convert the input string into an integer
number = Integer.parseInt(input);
```

4. Specific HTML String messages that you need to display on different game stages/ occasions: (**<p>** is a tag in HTML for new paragraph)

On game start:

```
Let's start!<p>
```

For each attempt:

```
Guess my number in [rangeStart, rangeEnd]<p>
n attempt(s) left<p>
```

Should the guess be out of the range:

```
Where are you kicking? Try again!<p>
```

Should the guess be within the range:

```
It's near, try again!<p>
```

Successful guess within 7 attempts:

```
Bingo! You've got it!<p>
```

Game over when all attempts are used up:

```
Sorry, game is over!<p>
```

5. If a guess falls within the range but missing the target, the program shrinks the range by the following rules:
   - If the guess is larger than the target, the new range is [previousRangeStart, guess – 1]
   - If the guess is smaller than the target, the new range is [guess + 1, previousRangeEnd]

6. On successful guess or game over, the program terminates with a MessageDialog, e.g.,

```
// pops up a message dialog in HTML; the first argument is usually null
JOptionPane.showMessageDialog(null, "<html>Line1<p>Line2<p>");
```

7. For debugging purpose, you may use **`System.out`** to print some diagnostic messages, say, the target number, attempt count, the current range, etc.

8. We assume the player cooperates so that no input validation or exception handling is required. For example, the user will NOT click Cancel button or Close the dialog abruptly; the user always inputs an integer on each attempt; etc.

9. Demonstration and sample program (in the form of a web-based applet):
   http://www.cse.cuhk.edu.hk/csci1530/asgdemo/GuessingDemo.html
   Your work should be a stand-alone Java application, not an applet.

Your Task:

1. Before you code, draw your own flow chart and think twice. Start with a simpler game, e.g., *a single-attempt number guessing game without any range. Build up incrementally.*

2. **Create** a new NetBeans project named **`Guessing`**, with a package named **`game`** and main class **`Guessing`**.

3. The main class should include a proper *header comment block*, similar to the one appeared in assignment 1. It should include course code and course name, title of the assignment, brief description of your work, your name, your SID, date of the work, as well as your statement of originality and declaration of understanding the guideline on academic honesty.

4. Zip and Submit your *whole NetBeans project folder* in an archive file **`Guessing.zip`** via our Online Assignment Collection Box on Blackboard <**`https://elearn.cuhk.edu.hk`**>

5. Enjoy your game!

Marking Scheme and Notes:

0. We will test playing your program vigorously for proper game features. So do you!

1. The submitted program should be free of any typing mistakes, compilation errors and warnings.

2. Comment/remark, indentation, style are under assessment in every programming assignments unless specified otherwise. Variable naming, proper indentation for code blocks and adequate comments are important.

3. Remember to do your submission before 6:00 p.m. of the due date. No late submission would be accepted.

4. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be counted. You may delete (i.e. take back) your attached file and re-submit. We ONLY take into account the last submission.

University Guideline for Plagiarism

Attention is drawn to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Details may be found at http://www.cuhk.edu.hk/policy/academichonesty/. With each assignment, students are required to submit a statement that they are aware of these policies, regulations, guidelines and procedures.

Faculty of Engineering Guidelines to Academic Honesty

MUST read: http://www.erg.cuhk.edu.hk/erg-intra/upload/documents/ENGG_Discipline.pdf (VPN required if you are not in the CUHK campus network)