/\*\* =======================================================================

\* Class:Lesson 43 Ex43.1-13 Pg.43.4 Author: Yin Linhai

\* Version:001Date:Feb 6, 2014

\*

\* Blue Pelican Lesson 43 questions

\*

\* Course:Computer Science 201Teacher:Mr Blakey

\* School:Sir Winston Churchill High School, Calgary, Alberta, Canada

\* Language: Java SE 7.0Target Operating System: Java Virtual Machine

\* System:Intel Celeron 3GHz running under Windows 7 IDE: Eclipse 4.2

\*========================================================================\*/

**package** exercises;

**public** **class** Lesson\_43 {

**public** **static** **void** main(String[] args) {

//Blue Pelican Lesson 43 Exercises

/\*Exercise 1

\*

\* ArrayList<String> alst = new ArrayList<String>();

\*/

/\*Exercise 2

\*

\* Objects

\*/

/\*Exercise 3

\*

\* It automatically scales, and has no fixed size

\*/

/\*Exercise 4

\*

\* If you use type parameters, then the ArrayList

\* will only take that type of data

\*/

/\*Exercise 5

\*

\* int x = 9

\* a.add(x);

\*/

/\*Exercise 6

\*

\* int gh = a.get(22);

\*/

/\*Exercise 7

\*

\* int sz = a.size();

\*/

/\*Exercise 8

\*

\* int kd = a.get(101);

\* a.set(17);

\*/

/\*Exercise 9

\*

\* a.add(59, 127);

\*/

/\*Exercise 10

\*

\* Empties the ArrayList

\*/

/\*Exercise 11

\*

\* alist.clear();

\*/

/\*Exercise 12

\*

\* String myString = buster.get(99);

\*/

/\*Exercise 13

\*

\* Object

\*/

}

}