

Program 3: An Instant-Win Lottery

CS 617 Winter 2013

1 Goals.

To define a complete class, write an entire program, use an array, and sort it.

2 Overview and Use Case

This program will simulate a very simple instant-win lottery game.

1. The gambler will pay \$1 for a ticket.
2. The computer will then calculate 5 numbers between 1 and 35 and store them in an array. After all five have been selected, use the `Arrays.sort` function to sort the array.
3. The gambler will select five numbers between 1 and 35. If he selects a number twice, prompt him to enter a different number.
4. The computer stores the selected numbers in an array, sorts them, and echo-prints them.
5. The computer then compares the two sets of numbers and counts how many match. It prints out the target numbers and the payoff.
6. Finally, the computer announces the result of this game.

3 Instructions

The Lottery Class. Define a class `Lottery` with five members:

- Two private data members, **winners** and **picks**, which are both arrays of ints.
- A public constructor with no parameters that will allocate memory for the two arrays.
- Two public functions, **play()** and **score()**.

play(). Prompt the user to put \$1 in the machine and click the “Play” button. (You can imagine that this is happening. You don’t need to do anything.) Then generate and store five random numbers between 1 and 35 (winners). Allow duplicates. Use `util.Arrays.sort()` to sort the specified array of ints into ascending numerical order. Do not print it yet. The prototype for sort is: `static void sort(int[] a)`

Ask the gambler to pick five numbers (**picks**) between 1 and 35 and validate the five inputs. Allow duplicates. Sort the array. Print it. When both arrays have been filled and sorted, call the **score()** function. It will return the number of matches. Then announce that the gambler has lost, or won, and announce the payoff \$3 for two matches, \$20 for three matches, \$100 for four matches, and \$500 if all five guesses match.

score(). The goal is to calculate the number of picks that match the winning numbers. Search the picks array for each winner. If you find a match, remove the number from the picks array by setting it to 0. When all five winners have been checked, return the number of matches (0’s zeros in the picks array).

The main function. Create a `Lottery` object, then call its **play()** function.

Turn in by Friday, January 25, 2013, if possible. Otherwise, come to class with a list of questions. Test your program until you get a loss and two different levels of wins. Use your mouse to copy and paste those outputs into a single file. Put the output file into your project directory, zip it all up, and email to my home.