Programming Fundamentals

Lab #4

Topics

- Creating classes and methods
- Instance, local, and static variables
- Creating static methods
- Accessing object/class methods and variables using the dot operator
- Arrays

Concepts

dot operator
return types (e.g. void, int, String, etc.)
static keyword
this variable
new keyword
toString() method
arrays

Exercise 1

In your **Box** class from the previous lab, add an overloaded method for ***printBox*** that takes 1 parameter: *char c*. This version should do the same as the ***printBox*** version with no parameters, except use the character *c* instead of *****.** Add code in the ***main*** method to invoke the second version of the ***printBox*** method and run it.

Exercise 2

Create a new class called ***Account*** with a main method that contains the following:

- A static variable called ***numAccounts***, initialized to 0.
- A constructor method that will add 1 to the ***numAccounts*** variable each time a new ***Account*** object is created.
- A static method called ***getNumAccounts*()**. It should return ***numAccounts***.

Test the functionality in the main method of ***Account*** by creating a few ***Account*** objects, then print out the number of accounts.

Exercise 3

Design and implement a class called Card that represents a standard playing card. Each card has a suit and a face value. Create a program that deals five random cards (with replacement). HINT: Use numbers to represent the suit and the face value and implement a toString method that returns a String corresponding to the given suit and face value numbers.

Exercise 4

Write a Java Class (**Numbers.java**) that contains a method called `nextLargest`. This method should accept an array of numbers and output, for each number in the array, the next bigger number. For example, if the array is

{78, 22, 56, 99, 12, 14, 17, 15, 1, 144, 37, 23, 47, 88, 3, 19}

the output should look like the following (? is a placeholder):

```
78: 88
22: 23
56: 78
99: 144
12: 14
14: 15
17: 19
15: 17
1: 3
144: 2147483647
37: 47
23: 37
47: 56
88: 99
3: 12
19: 22
```

NOTE 1: If there is no bigger number in the sequence, just display the value of Integer.MAX_VALUE .

NOTE 2: ? should be replaced with the appropriate number

Test the method by creating an array and calling it from the main method.