# Design, Exam, and Polymorphism – Welcome Back!

## Task 3: Exam

1. One question gave people more trouble that I thought it should. Given the following method header, list ALL of the different states that the array could be in that you must account for in your code. Do not write any code but think through what should have been handled.

/\*\* getSmallest returns the smallest value among the temperature values in the array.

If the lowest cannot be calculated from the array, return -999

 @param temperature A 2-D array of temperatures

 @return The lowest temperature in the array (or -999 if none found)

\*/

1. public double getSmallest ( double [][] temperature) If I declare two objects as follows, and Shape is the superclass and Circle is the subclass, which instantiation is legal (will compile)?

Shape blob;

Circle round;

* 1. blob = new Circle();
	2. round = new Shape();
1. Review the exam with one another to get correct answers. See the instructor if you have any question about what is the correct answer.

Task 4: Design issues – Review of PA3. These are questions for each individual. No group response is needed.

There were a lot of design issues that I saw in PA3. Please review your own PA and see if any of these pertain to you. Then share each question with the rest of the group. See who is designing well, and who could use some help:

1. Did you use constants? 21 and 15 had meaning in the context of this program. They should have been labeled. Where would you in your code use constants? Would you have used public constants? Why or why not?
2. Every class should have had a constructor (except for some of the Suit classes). The constructor would be used to initialize the attributes. Did you use a constructor in each class, particularly those with instance variables?
3. Did you pass parameters and return values from the majority of your methods in the main game class? Or did you build a bunch of attributes to save you the trouble? It is a better practice to keep local those values which do not directly describe the state of the object. Look at someone in your group that used only a few attributes in the game class. How did they solve the problems of using multiple methods?
4. How did you build your deck?
5. Did your shuffle method (if you had one) rebuild the deck if necessary or did you force the user of the class to do things twice?
6. How did you solve the problems of:
	1. shuffling the deck?
	2. preventing duplicate cards?
	3. calculating points?
	4. calculating points in Hand? when the player had one ace? two aces? Did you implement calculate points solely in Hand (hint: it should have been). What did your teammates do?
7. Finally, share with your group the one thing that gave you the most trouble and see if they have any better ways of solving that problem.