

**Due Date: Thursday May 1<sup>st</sup> 2008 before 11PM (to be submitted online via SPARK)**

**Homework 6:**

1. How does polymorphism differ from inheritance?
2. What do interfaces allow a programmer to do that abstract classes do not? If you cannot think of anything, consider trying to use many interfaces at once and many abstract classes at once.
3. List three or four differences between abstract and non-abstract methods.
4. Write a third class that implements the interface *number* given in the lecture 23 materials on the Syllabus web page. This third class can be anything you want besides the two classes given in lecture that already implements *number*. Modify the *main* class, also given in lecture, to include your new class for values of  $i > 10$ ; modify the *for* loop so that it ends when  $i == 15$ , and print the numbers 10 to 14 using your new class. Show a screenshot of the *main* class working.
5. Assume that a child class has overridden one of its parent's methods. If a reference refers to an instance of the child class but the reference variable is of the type of the parent class, and if this polymorphic reference calls the method, which version will Java recognize? How is this advantageous to programmers?
6. Write a recursive method that determines the minimum value of an array of 10 integer numbers. The array must be initialized to 10 random integer numbers. Show a screenshot of your method working.