```
// Bellevue College CS211
// Fall 2012, Exam 1 (100 pts)
// W.P. Iverson, instructor                NAME: W.P. Iverson

public static void main(String[] args) {
// The following program DOES run, only because I comment out lines numbered 9 to 14.
// Lines 1 to 8, you indicate the output, these are 5 points each, first two are easy.

int i = 42;
Integer j = new Integer(12);  // 1. (is there output?) NO
Integer k = new Integer(42);  // 2.  NO output


System.out.println(i < 42);         // 3. (true or false?) F
System.out.println(i == 42);        // 4.            T


System.out.println(j.toString());          // 5. 12
System.out.println(j == k);                // 6. F


System.out.println(j.compareTo(13));       // 7. -1
System.out.println(j.compareTo(k));        // 8. -1


// The following will run fine, if you follow instructions below, then lines
// 9 to 14 could have // removed to reveal the correct answers (10 points each)

// Now we'll build a couple of objects, called first and second below
// Both Classes for these objects are incomplete (see attached)
// You get to complete the Class programming here to produce the following output
// Please write constructors that preload three Integers [8, 2, 5] upon instantiation
// HASArrayList first = new HASArrayList();
// 9. Write the default constructor on attached page
//    Be certain to included code to add these data      (see next page)

// ISArrayList second = new ISArrayList();
//10. Write the default constructor on attached page
//    Again, use add() to get these data loaded

// In problems 11. and 12., you decide what method(s) are be needed to get these to work
// And write that code on attached Class pages.
// CAREFUL: You cannot just concatenate ", " on each Integer, the last one is different!
// System.out.println(first);  // 11. Output needs to be [8, 2, 5]
// System.out.println(second); // 12. Output needs to be [8, 2, 5]




// Separate issue here:  Do you understand the Comparable interface?
// Integer one = first.get(0); - n/a ?
// Integer two = second.get(1);

// System.out.println(one.compareTo(two));      // 13. Output is -1, 0, or 1 ???  +1
// System.out.println(two.compareTo(one));      // 14. Output is -1, 0, or 1 ???  -1
// This class "is a" ArrayList via inheritance
public class ISArrayList extends ArrayList<Integer> {

    // #10
    public ~~static~~ ISArrayList() {
        super();
        add(8); add(2); add(5);
    }

    // #12 toString alreday inherited

}
```

```
// This class "has a" ArrayList (import java.util.ArrayList) as a member
public class HASArrayList {
      private ArrayList<Integer> data;

    #19
    public HASArrayList() {
        data = new ArrayList<Integer>();
        data.add(8); data.add(2); data.add(5);

    #11
    public String toString() {
        return data.toString();
    }

    #13
    public Integer get(int i) {
        return data.get(i);
    }
}
```