CS 61B: Data Structures (Spring 2012)
Midterm I

If you want to relive taking the midterm, here it is in PostScript or PDF.

Solutions

Problem 1. (8 points) **A miscellany.**

a. Binary search begins by inspecting the middle element of a list or sublist thereof. If the list is linked, you will have to traverse half the nodes in the list/sublist to reach the middle one. In general, binary search takes at least as long to reach any given node as a naive search.

b. The compiler prints out the class, method, and line number where the exception occurred. Search that line for a "dot" (dereference) operation.

c. These lines cannot throw run-time errors.

```
boolean b1 = (head != null) && (head.next != null);
boolean b2 = (head == null) || (head.next == null);
Object o1 = (java.io.InputStream) System.in;
int ia[] = {3, 7};
```

d. `public final`. (It is acceptable to replace `public` with the same protection level as the array itself has, though it makes no difference.)

e. A cast changes the static type of an expression, and its effect on dynamic method lookup is nothing whatsoever.

Problem 2. (10 points) **Inheritance.**

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
public class Rational implements Comparable {
  public int numer;     // Numerator of a fraction (rational number).
  public int denom;     // Denominator.  Invariant:  denom is never zero.
  private static short compare;

  public int compareTo(Object o) {
    Rational other = (Rational) o;
    long myProduct = (long) numer * (long) other.denom;
    long otherProduct = (long) other.numer * (long) denom;
    if (myProduct == otherProduct) {
      compare = 0;
    } else if (myProduct < otherProduct ^ denom < 0 ^ other.denom < 0) {
      compare = -1;
    } else {
      compare = 1;
    }
    return compare;
  }

  public Rational(int n, int denom) {
    numer = n;
    this.denom = denom;
    if (denom == 0) System.exit(1);      // Enforce the invariant.
  }

  protected static short lastCompareTo() {
    return compare;
  }
}

public class Integr extends Rational {
  public Integr(int n) {                 // Construct the integer n/1.
    super(n, 1);
  }
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
  public int compareTo(Object o) {
    if (o instanceof Integr) {
      if (numer < ((Integr) o).numer) {
        return -1;
      } else if (numer == ((Integr) o).numer) {
        return 0;
      } else {
        return 1;
      }
    }
    return super.compareTo(o);
  }
}

public class WholeNum extends Integr {   // Invariant:  numer is never zero.
  public WholeNum(int n) {               // Construct the whole number n/1.
    super(n);                            // super(n, 1); is also correct.
    if (n == 0) System.exit(1);          // Enforce the invariant.
  }

  public short makesNoSense() {
    return Integr.lastCompareTo();
  }
}
```

Problem 3. (7 points) **Run-length encoding an array of strings.**

```
  public SListNode rle(String[] strings) {      // Run-length encode an array.
    int i = strings.length - 1;
    SListNode head = new SListNode(strings[i], 1, null);
    for (i--; i >= 0; i--) {
      if (strings[i].equals(head.item)) {
        head.count++;
      } else {
        head = new SListNode(strings[i], 1, head);
      }
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    }
    return head;
  }
```

Mail inquiries to [cs61b@cory.eecs]

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com