Review for Final

There’s a lot more to CS than programming ...
But there isn’t a lot more to the CIS 210 final exam
What to expect

Fewer “what does this print” questions
  • And maybe harder ... e.g., they could include recursion or more complex loops

More “write a function/method/class to do X”
  • Mostly Java ... I won’t require Python, but may permit Python for some questions
Scope

Everything you know ... but mostly
- Methods/functions
- Algorithm design
  - loops, conditions, etc
- Classes and objects

Focus on problem-solving logic more than syntax
Loop idioms

On collections:

Accumulate  e.g., sum, product
Select  e.g., largest, smallest
Filter  e.g., remove spaces
Do something to each  e.g., double each
Scan  e.g., count words

A typical exam question asks you to do these in some combination, such as select + accumulate
Accumulate

Typical form: Summing an array

```java
sum = 0;
for (int i=0; i<coll.length; i++) {
    sum = sum + coll[i];
}
```
Accumulate: With iterable collection

Typical form: Summing an ArrayList<Typ> list

```java
sum = 0;
for (Typ x: list) {
    sum = sum + x;
}
```
Select an extreme

int[ ] ar;
if (ar.length < 1) { return default; }
int max = ar[0]; // biggest or smallest or greenest or ... 
for (int i=0; i < ar.length; ++i) {
    if (ar[i] > max) { // or smaller, or greener or ...
        max = ar[i];
    }
}
return max;
Select an extreme

ArrayList<T> ar;
if (ar.size() < 1) { return default; }
int max = ar.get(0);
for (T elem: ar) {
    if (elem > max) {
        max = elem;
    }
}
return max;
Extreme selection from midterm redux

/**
 * Determine the maximum "step" from one element
 * of an array to the next element. For example, if the
 * input array is [ 1 3 2 7 4 8 ], then the maximum
 * step is 5 (from 2 to 7). (Note that steps can be
 * positive or negative. 2 is a larger step
 * than -3, and -3 is a larger step than -5.)
 * Special case:
 * If the array has less than 2 elements, then maximum
 * positive step is zero.
 * @param ar  An array of integers.
 * @return The maximum x such that x == a[i+1] - a[i],
 * or 0 if length of ar is less than 2.
 */
public static int maxStep(int [ ] ar) {

}
public static int maxStep(int[] ar) {
    if (ar.length < 2) { return 0; }
    int max = ar[1] - ar[0];
    for (int i=2; i < ar.length; ++i) {
        int step = ar[i] - ar[i-1]; // scan idiom
        if (step > max) { // extreme idiom
            max = step;
        }
    }
    return max;
}
Normalize (extreme + “do to each”)

I nearly put this on the midterm redux ...

Given an array \( ar \) of double,
replace each value by a value \( x \)
between 0.0 and 1.0 such that
if min and max are the largest and smallest
values in \( ar \),
the old value is \( \text{min} + x(\text{max} − \text{min}) \)
What about classes and objects?

You should understand

• How to create an object from a class (with “new”)
• The difference between static and non-static methods, and how to call each
• The difference between variables that are local to a method, and variables that are local to an object, and how to use both
• That a variable of an object type is really a reference, and the difference between references two references to the same object and two references to objects of the same class
Logistics

Both Greg Bickerman and I will hold help sessions during finals week

When? We need your suggestions.

Final exam is 2 hours, 15:15 to 17:15, Thursday

Open book, open notes ... no computers, phones, etc.