CMSC 433 midterm review

Oct 23rd, 2007
Check out old 433 exams

- Follow exams link from course web page
- previous offerings don't cover the same material
  - if you see a question on material we didn't cover, ignore it
    - it isn't relevant to your midterm preparation
Testing

- JUnit
- Mock Objects
- Code coverage
  - statement, branch coverage
TCP Sockets, HTTP

- ServerSockets, Sockets
- HTTP
Generics

given a non-generic Stack class:
  
  - class Stack {
    int top = -1;
    Object a[];
    Stack(int maxCap) {
      a = new Object[maxCap];
    }
    Object pop() { Object r = a[top];
      a[top--] = null; return r; }
    void push(Object e) { a[++top] = e; }
  }

  • modify it to be a Stack<E>
Bounded Wildcards in Generics

• What is the difference between defining something as:
  – Collection<?> c
  – Collection<Object> c
  – Collection<? extends E>
  – Collection<? super E>

• Where/when might these be used (e.g., method parameters, return values)

• See "Effective Java Reloaded" slides
Bounded type variables

• If you need a type that is bound and occurs in more than one place:

<T extends Number>
    T sum(List<T> x) { ... }

Source code version control
Design patterns

- Observer
- Decorator
- Adapter
- Visitor
Concurrency and Synchronization

- synchronized
- wait/notifyAll

- java.util.concurrent.
  - ReentrantLock
  - Lock (lock, unlock, tryLock, ...)
  - Condition (await, signal, signalAll)
Concurrency bugs

- Data races
- atomicity failures
- Deadlock
Guarding against concurrency bugs

• Documenting concurrency
  – GuardedBy annotation
• Immutability