CSPP 511-01:
Introduction to Object-Oriented Programming

Harri Hakula

Ryerson 256, tel. 773-702-8584
hhakula@cs.uchicago.edu

August 9, 2000
Outline

• More on Exceptions
• Recursion
• Project
More on Exceptions

Sometimes it is desirable to guarantee that part of program is executed whether an exception was thrown or not. This is accomplished with the finally-block.
try-throw-catch-finally

{ ...
  try {
    doSomething();
    if(error)
      throw new Exception("I’m dying...");
    else doMore();
  } catch(Exception e) { ... }
  finally { ... } // Always executed
}
finally

In the previous example, the `finally`-block is always executed. This leads to so-called pre-post-idiom:

Object val = null;
try {
    val = pre();
    // other actions
} finally {
    if(val != null)
        post();
}
Recursion

If a method definition contains an invocation of the very method being defined, then that invocation is called a recursive call or recursive invocation.

```java
public void printd(int n) {
    if(n < 0) {
        n = -n;
        System.out.print("-");
    }
    if(n / 10 > 0) printd(n / 10);
    System.out.print("" + n % 10);
}
```
Recursion Cntd.

Every recursive algorithm must have three steps:

1. End condition: When does the recursion end?
3. Recursive step: Selection of the next argument.
public class ArraySearcher {
    public ArraySearcher(int[] theArray) {
        a = theArray;
    }
    public int find(int target) {
        return search(target, 0, a.length - 1);
    }
    private int search(int target, int first, int last) {
        int result = -1;
        int mid;
        if (first > last)
result = -1;
else {
    mid = (first + last)/2;
    if (target == a[mid])
        result = mid;
    else if (target < a[mid])
        result = search(target, first, mid - 1);
    else /*(target > a[mid])*/
        result = search(target, mid + 1, last);
}
return result;

private int[] a;
When Should One Iterate, When Recurse?

To iterate is human, to recurse, divine. –Anonymous.
Project