Overview

- Homework #1
  - Due next Friday.
  - See the web page.
- An implementation of Polynomials

An Implementation of Polynomials

- Create
- Evaluate
- Add
- Print

Controlling Access to Members of a Class

Specifier	class	subclass	world
private	X		
protected	X	X	
public	X	X	X

Private

- Can one object access the private members of another object of the same class? *Yes.*

- Objects of the same type have access to one another's private members. This is because access restrictions apply at the class or type level (all instances of a class) rather than at the object level (this particular instance of a class).

Private (cont.)

- Example

```
class Alpha {
  private int iamprivate;
  boolean isEqualTo(Alpha anotherAlpha) {
    if (iamprivate == anotherAlpha.iamprivate)
      return true;
    else
      return false;
  }
}
```

"this"

- In any method, `this` refers to the object on which the method is being performed. It's useful when you...

  1. need to access a field that is obscured by a parameter or
  2. want to pass the object as an argument to a method.

“this” (cont.)

- Example

```
class Point2D {
    double x, y;

    Point2D(double x, double y)
    {
        this.x = x;
        this.y = y;
    }
```

Exceptions

- An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions.
- Many kinds of errors can cause exceptions
  - Hardware error
  - Programming error: dereferencing `null`
- When such an error occurs within a Java method, the method creates an exception object, which describes the exception, and hands it off to the runtime system, which is responsible for finding code to handle the error.

Exceptions (cont.)

- Where does the runtime system look?
- The run-time system searches backwards through the chain of method calls, beginning with the method in which the error occurred, until it finds a method that contains an appropriate exception handler.
- What is an appropriate exception handler?
- The type of the exception is the same as the type of exception handled by the handler.

Exceptions (cont.)

- What are the advantages?
  - Separates error handling code from regular code.
  - Automatically propagates errors up the chain of method calls.
  - Groups error types and differentiates errors.

Catching Exceptions

An arbitrary number of `catch` statements can follow the `try` statement.

```
try {
    ...
    methodThrowsExcType12Or3();
    neverCalledMethod();
    ...
} catch (ExcType1 e) {
    ...
} catch (ExcType2 e) {
    ...
} catch (ExcType3 e) {
    ...
```