UNIVERSITY OF CALGARY
SCHULICH
School of Engineering

ENSF 593/594

9 - Java Interfaces

Interfaces

- Interfaces are unimplemented declarations of methods and/or constants.

Interfaces Example

```
public interface Color {
    static final int RED = 0;
    static final int BLUE = 1;
    static final int YELLOW = 2;

    public void setColor(int color);
    public int getColor();
}
```

constants

methods (implicitly abstract)

Interfaces (continued)

- Any class can implement an interface.
  - Use the “implements” keyword.
  - Provide code for *all* methods in the interface.

Interfaces Example

```
public class Point extends Object implements Color
{
    private double x, y;
    private int color = RED;
    …
    public void setColor(color) {
        this.color = color;
    }

    public int getColor() {
        return color;
    }
}
```

this class implements the Color interface

can use constants from the interface

actual implementation of the interface

Interfaces (continued)

- A class can implement more than one interface. E.g.

```
public class Point extends Object
        implements Color, Cloneable
{
    ...
}
```

add interface names after the implements keyword, separated by commas

Interfaces (continued)

- Interfaces give Java some of the power of multiple inheritance, without any of its problems.
  - However, there is no code reuse, since each class must re-implement the methods.
- An interface is normally put into its own Java source file. E.g. `Color.java`

Interfaces (continued)

- Interfaces can extend one or more super-interfaces.
  - A sub-interface inherits all the constants and method declarations from the super-interfaces, and may add its own.
- Example:

```
public interface Measurable extends
        Weight, Volume, Velocity {
    // additional method declarations here
}
```

Interface (continued)

- A reference of an interface type can refer to the instances of any classes that implement that interface:

Example

Class Exercise

Scenario

- As part of word processing software, you need to develop a few class such as Text, Shape, Document, etc..
  - Class Document uses the other two classes
  - Class Text and Shape Implements a Java interface called Resizeable, allowing the shapes and text to enlarged or shrunk.

Class Exercise

Question:

- Why is an interface needed?