Function Objects and Comparators

CSSE 221

Fundamentals of Software Development Honors

Rose-Hulman Institute of Technology

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Comparator vs. Comparable

- Comparable: implemented within the object you want to compare

- Comparator: implemented within a class that compares objects

Comparable

- If a class implements comparable interface, then it is required to provide a compareTo() method

- When you compare objects, the compareTo() method returns #<0, 0, or #>0

For Example

```
public class Coin implements Comparable<Coin>
{
      -Class variables and methods

      public int compareTo(Coin other)
      {
        if(this.getValue()>other.getValue())
            return 1;
        if(this.getValue()<other.getValue())
            return -1;
       return 0;
      }
 }
```

For Example

```
public class Coin implements Comparable<Coin>
{
    -Class variables and methods

    public int compareTo(Coin other)
    {
        return(this.getValue()-other.getValue());
    }
 }
```

Comparator

- Comparator is used nearly the same as Comparable, but requires a separate class that compares two objects passed to the Compare() method.

Comparator (cont.)

- The real reason to do this is to use
  - Collections.sort(List<Object>, Comparator<Object>);
  - This is necessary when the objects stored in the list do not implement Comparable, and there is no way to make the object implement Comparable.

For Example

```
public class compareCoins implements Comparator<Coin>
{
    public int compare(Coin a, Coin b)
        {
          return a.getValue()-b.getValue();
        }
}
```

For Example

```
Public class CoinTester
{
public static void main(String[ ]args)
        {
                //create a list of coins.
                Collections.sort(list, new compareCoins());
        }
}
```

Function Objects

- These are only used to provide functionality for a method or library
- Small class only meant to have one or two methods, basically an Interface.

Activity Time

- You are going to compare yourself to every person… based on height.
- Tally up your 1's, 0's and -1's.
- Then Sort yourselves, only based on the 1,0,-1 information.

Demo Time

- Checkout ComparatorsSection2 from your SVN public repository
- Write a Comparator class to sort car.java

Your Turn

- Write the compareTo() method in Rainbow.java