



CMSC 131

Object-Oriented Programming I

Review II

Dept of Computer Science
University of Maryland College Park

This material is based on material provided by Ben Bederson, Bonnie Dorr,
Fawzi Emad, David Mount, Jan Plane

Check Your Grades

- First this is not about submitting regrade requests or address grading concerns about material already graded and for which we already posted a deadline that has passed. This is about double-checking any corrections you already submitted and to make sure everything is in order
- We will not address any grading concerns the day of the final exam
- If you see any errors in your scores contact your teaching TAs immediately

ArrayList

- Resizable array
- Example of generic class
 - `ArrayList<String>`
 - `ArrayList<Integer>`
 - Cannot have `ArrayList<int>`
- Can be used with for each

Interface

- Allow us to enforce implementation of methods
- Defines an IS-A relationship (similar to extending a class)
- Can we have an interface without any methods?
 - How can it help us?
- Comparable interface
 - Defines compareTo Method
 - Which classes implement it?
- Polymorphism
 - Using an interface we can create one variable that can reference objects different types
 - We can also use a superclass like we have done with the Person/Student inheritance hierarchy
- **Example:** Vehicle.java, Bike.java, Automobile.java, Driver.java

Inheritance

- What is the super/base class?
- What is a sub/derived class?
- What is super?
 - **super** can be used to call super class constructors
 - **super** can be used to call super class methods
 - Can **super** and **this** (to call constructors) be used at the same time in a method?
- Why we want to override a method?
- Why we want to overload?
- How can you tell whether you are overriding or overloading a method?
- What is the correct way to implement equals?
- Why would you need to downcast?
- Can we have multiple inheritance in Java?
- **Example:** Automobile.java, RaceCar.java, Driver2.java