6. Interfaces :: Example Program

```
// MakesSound.java
public interface MakesSound {
  public void makeSound();
}

// Mammal.java
public abstract class Mammal implements MakeSound {
  // Note that Mammal does not implement MakesSound.makeSound().
}

// Insect.java
public abstract class Insect implements MakesSound {
  // Note that Insect does not implement MakesSound.makeSound().
}

// Dog.java
public class Dog extends Mammal {    // A Dog is a Mammal.
  @Override                          // Dog is overriding makeSound() inherited from Mammal.
  public void makeSound() {
    System.out.println("Bark");      // Dogs bark.
  }
}
```

6. Interfaces :: Example Program (continued)

```
// Cat.java
public class Cat extends Mammal {    // A Cat is a Mammal.
  @Override                          // Cat is overriding makeSound() inherited from Mammal.
  public void makeSound() {
    System.out.println("Meow");      // Cats meow.
  }
}

// Cricket.java
public class Cricket extends Insect {  // A Cricket is an Insect.
  @Override                            // Cricket is overriding makeSound() inherited from Insect
  public void makeSound() {
    System.out.println("Chirp");       // Crickets chirp.
  }
}

//Main.java
import java.util.ArrayList;

public class Main {
  public static void main(String[] args) { new Main().run(); }
```

6. Interfaces :: Example Program (continued)

```
  public void run() {
    // critters is an ArrayList of various sound-making critters.
    ArrayList<MakesSound> critters = new ArrayList<>();
    critters.add(new Dog());
    critters.add(new Cat());
    critters.add(new Cricket());
    critters.add(new Cat());
    critters.add(new Cricket());
    beNoisy(critters);
  }

  public void beNoisy(ArrayList<MakesSound> pCritters) {
    for (Makessound critter : pCritters) {
      critter.makeSound();
    }
  }
}
```

Output

```
Bark
Meow
Chirp
Meow
Chirp
```