1

Handout A for e02 CS56 F17

Beverage.java

```
 1  public class Beverage extends Product implements Edible {
 2
 3      private int calories;
 4      private double fluidOunces;
 5
 6      public Beverage(int price, String name,
 7                  int calories, double fluidOunces) {
 8          super(price, name);
 9          this.calories = calories;
10          this.fluidOunces = fluidOunces;
11      }
12
13      public int getCalories() {return this.calories;}
14      public double getFluidOunces() {return this.fluidOunces;}
15  }
```

Edible.java

```
1  /** something that can be eaten */
2  public interface Edible {
3      public int getCalories();
4  }
```

Food.java

```
 1  public class Food extends Product implements Edible {
 2
 3      private int calories;
 4      private double weight;
 5
 6      public Food(int price, String name,
 7                  int calories, double weight) {
 8          super(price, name);
 9          this.calories = calories;
10          this.weight = weight;
11      }
12
13      public int getCalories() {return this.calories;}
14      public double getWeight() {return this.weight;}
15  }
```

FreeCandy.java

```
 1  public class FreeCandy implements Edible {
 2
 3      private int calories;
 4
 5      public FreeCandy(int calories) {
 6          this.calories = calories;
 7      }
 8
 9      public int getCalories() {return this.calories;}
10  }
```

Product.java

```
 1  public abstract class Product {
 2      String name;
 3      int price;
 4
 5      public int getPrice() { return price; }
 6      public String getName() {return name;}
 7
 8      public Product(int price, String name) {
 9          this.price = price;
10          this.name = name;
11      }
12  }
```

End of Handout

Code for

`PartialFoods` problem