CSC 130 | Fall 2017 | Sample Quiz Questions

Find the error

Starting with the next page you will see a sequence of problems that appear very similar. On the left will be an implementation of a `Pizza` class. There will be an error in some section of code.

Without comparing the pages to each other, take 3 steps in each **individual** page:

1. identify the error (circle the bad code)
2. explain the error (why is the code bad)
3. suggest new code that fixes the error

Note that in some cases the "error" consists of code that is syntactically correct but is poorly designed or produces a result that does not make any sense given the context of the problem.

```java
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public int getSizeCentimeters() {
    diameter = (int)(diameter * 2.54);
    return diameter;
  }
}
```

Accessors cannot change instance variable values

```java
  public int getSizeCentimeters() {
    int cm = (int)(diameter * 2.54);
    return cm;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int diameter) {
    toppings = new ArrayList<String>();
    diameter = diameter;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Do not use same name for input parameter and instance variable

```
  public Pizza(int diam) {
    toppings = new ArrayList<String>();
    diameter = diam;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = "";
    diameter = 0;
  }

  public Pizza(String foods) {
    toppings = foods;
    diameter = 0;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

toppings is an ArrayList<String> not a String

```
  public Pizza() {
    toppings = new ArrayList<String>;
    diameter = 0;
  }

  public Pizza(ArrayList<String> foods) {
    toppings = foods;
    diameter = 0;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public void getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Accessor methods cannot have a void return type

```
  public int getSize() {
    return diameter;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int diameter) {
    diameter = diameter;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Do not name input parameters identically to instance variables

```
  public void setSize(int newSize) {
    diameter = newSize;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public void Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public void Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Constructors do not have *any* return type

```
  // public void Pizza() {
  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }
  // public void Pizza(int size) {
  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public int setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Mutators should not return values

```
  public void setSize(int newSize) {
    diameter = newSize;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public String getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

toppings is an ArrayList, not a String

```
  public ArrayList<String> getToppings() {
    return toppings;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  ArrayList<String> toppings;
  int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Instance variables should be private

```
  private ArrayList<String> toppings;
  private int diameter; // in inches
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    size = diameter;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Assign to the instance variable, not the parameter

```
  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }
```

```java
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return toppings.size();
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Returns the wrong data

```java
  public int getSize() {
    return diameter;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public new Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public new Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

new is used to create objects, not to define object constructors

```
  // public new Pizza() {
  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }
  // public new Pizza(int size) {
  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }
```

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    diameter = 0;
  }

  public Pizza(int size) {
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

You must initialize all instance variables in each constructor

```
  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }
```

```java
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    newSize = diameter;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```

Set the instance variable, not a parameter

```java
  public void setSize(int newSize) {
    diameter = newSize;
  }
```

This is a correctly implemented class with no errors.

```
import java.util.ArrayList;
public class Pizza {

  private ArrayList<String> toppings;
  private int diameter; // in inches

  public Pizza() {
    toppings = new ArrayList<String>();
    diameter = 0;
  }

  public Pizza(int size) {
    toppings = new ArrayList<String>();
    diameter = size;
  }

  public void add(String topping) {
    topping = topping.toLowerCase();
    toppings.add(topping);
  }

  public void remove(String topping) {
    for (int i=toppings.size()-1 ; i>-1; i--) {
      String t = toppings.get(i);
      if (t.equals(topping)) {
        toppings.remove(i);
      }
    }
  }

  public ArrayList<String> getToppings() {
    return toppings;
  }

  public boolean hasTopping(String topping) {
    for (String t : toppings) {
      if (t.equals(topping)) {
        return true;
      }
    }
    return false;
  }

  public void setSize(int newSize) {
    diameter = newSize;
  }

  public int getSize() {
    return diameter;
  }

  public double getSizeCentimeters() {
    double cm = diameter * 2.54;
    return cm;
  }
}
```