Static Scope

Ginger Bread Men Race

- When I think of the Java Static property, I think of a Race, run by Ginger Bread Men

Cookie Cutter Class

- To make the race interesting, we probably need a bunch of ginger bread men. To do this, we will get a cookie cutter to make lots of ginger bread men. In Java, this looks like:

```
Public class CookieCutter {
}
```

- The Cookie Cutter is a form for making cookies; but it is made of metal; so you wouldn't want to eat it like a cookie. Just make more cookies with it.

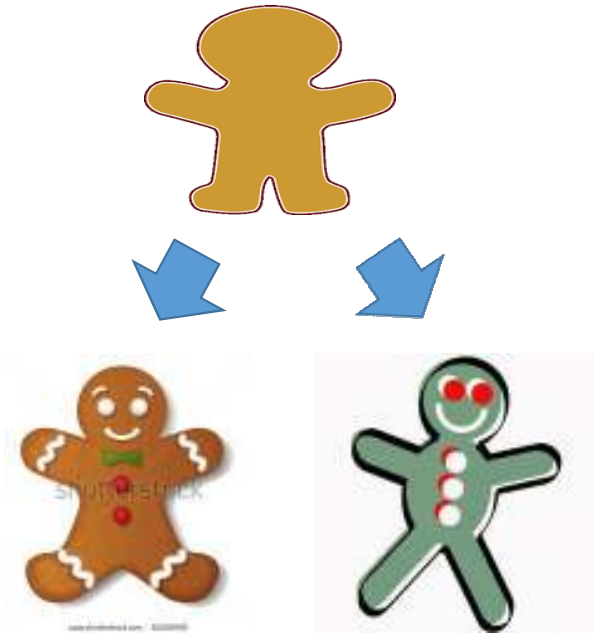
Cookie Cutter Features

- The cookie cutter just gives a general shape to the cookie.
- It can have general features for the cookie, like eye color or number of buttons; but these will be blank in the cookie cutter, and will be customized later.
- The cookie cutter can also define some things the cookie can do, like walk or run. In Java, this looks like:

```
public class CookieCutter {
   String eyeColor;
   int numberOfButtons;
   String runWalk;
   CookieCutter (String eyeColorIn, int numberOfButtonsIn) {
      eyeColor = eyeColorIn;
      numberOfButtons = numberOfButtonsIn;
   }
   public void walk() {
      runWalk = "Walking";
   }
   public void run() {
      runWalk = "Running";
   }
}
```

Creating a Cookie (Instantiation)

- When we use the cookie cutter to make a cookie, it is blank
- After we create the cookie, we can customize it with eye color, or number of buttons
- In Java, this looks like:
- CookieCutter george = new CookieCutter(“White”, 2); - which creates a white eyed, 2-buttoned cookie
- CookieCutter sarah = new CookieCutter(“Red”, 3); - which creates a red eyed, 3-button cookie



Cookies are Dynamic

- Cookies are dynamic. They can move around on the race course, from the start to the finish. You can even change whether they are running or walking. In Java:

  george.running();

  sarah.walking();

  without affecting any of the other cookies
- You can change the properties of a cookie, like changing the eyes from red to blue, or one of the buttons might fall off

  sarah.eyeColor = “blue”;

  george.numButtons = 2;
- You can make as many cookies as you want

Some things are Static

- However, some things you only want one of, and you don't want them to move. Like the Start or Finish line of the race. You wouldn't want three start lines, that would be confusing, and you wouldn't want the finish line to move around, that would be crazy. So, some things don't change, and they are labelled with the property static. In Java:

    public static main(…)

(Note: in Java, the main function is the starting line of the Java program. You only want one of these, and you don't want it to move around).

- There are also some bits of information that you want everyone to be able to go to one fixed location to find. For instance, ginger bread men are very concerned about whether it is raining or not. So, we might want one variable that everyone can look at to see if it is raining.

    public static boolean isItRaining = false;

Static vs Dynamic

- Static
- Needs to be specified with static keyword
- There can only be one of these
- This is in a fixed location
- If this is public, everyone can locate this

- Dynamic
- Is the default unless it is static
- There can be several instances of these items
- This can be in different locations in memory
- Even if this is public, you need to know the location of the instance to be able to access this