



CSE 115/503
March 28 – April 1, 2011

Announcements

- Lab 6 continued this week in recitation
- Grades on UBLearns
 - Exam grade recorded incorrectly – need to bring the exam to me (IN PERSON) preferably during office hours
 - Lab grade question – email Ming
 - Lab grade recorded incorrectly – can email me with the Lab and which section you submit to
 - Other questions – office hours
- Resign deadline is Friday, April 1st
- Exam 4 is Monday, April 4th

Getting all the shapes

- Look at DrawingCanvas class documentation

Method Summary	
<code>void add(IGraphic g)</code>	Adds a Graphic onto the DrawingCanvas
<code>Collection<IGraphic> getAllGraphics()</code>	Gets a Collection of all the Graphics on this DrawingCanvas
<code>Point getCenterLocation()</code>	Gets the center of the DrawingCanvas
<code>void mouseClicked(MouseEvent e)</code>	Called when the mouse is clicked on the DrawingCanvas, delegates to the appropriate Graphic
<code>void mouseDragged(MouseEvent e)</code>	
<code>void mouseEntered(MouseEvent e)</code>	Called when the mouse is entered onto the DrawingCanvas, does

getAllGraphics method

- Returns

`Collection<IGraphic>`

Question 2: What do you think a Collection is?

Collection

- In the context of this method:
- Collection is a `java.util.Collection`, an interface in the `java.util` package.
- However, in general, the idea of a collection is a bigger concept. A collection is a group of things.

Why do we need them?

- Variables
- Have a value
- One value
- What if we want to keep track of multiple things?

Another Question

Collection<IGraphic>

So what is this part?

It's a type

- It's the type of thing inside the collection.
- Formally, <IGraphic> is called a generic.
- Ironically, *generics specify* the type of "thing" a collection holds.
- It also provides compile-time checking of the contents of the collection.

Example

```
Collection<Cat> bagOfCats =  
    something.getAllCats();
```

```
bagOfCats.add(new Cat());
```

```
bagOfCats.add(new TabbyCat());
```

```
bagOfCats.add(new StrayCat());
```

```
bagOfCats.add(new Dog());
```

Problem!

- The compiler would not allow this at compile time.

Our example

- Question 3
- Write the code to create the local variable and assign the collection to it.

What's next

- We need to go through the "bag" and tell each graphic to move up.

How do we do that?

- Get a graphic out of the bag
- Move the graphic up
- Repeat steps 1 & 2 until we moved all the graphics up on the screen.

for-each loop

```
for(TypeOfElementInCollection identifier: referenceToCollection)  
{  
    // code that indicates what to do for each element in the  
    // collection  
}
```

Note that in the first part of this, we are actually declaring a variable. We will use the variable in the body of the loop to “do something” with each of the elements that are stored in the collection.

The reference to the collection is typically the name of the variable that refers to the collection.

```
for(Cat c: bagOfCats) {  
    c.pullTail();  
}
```