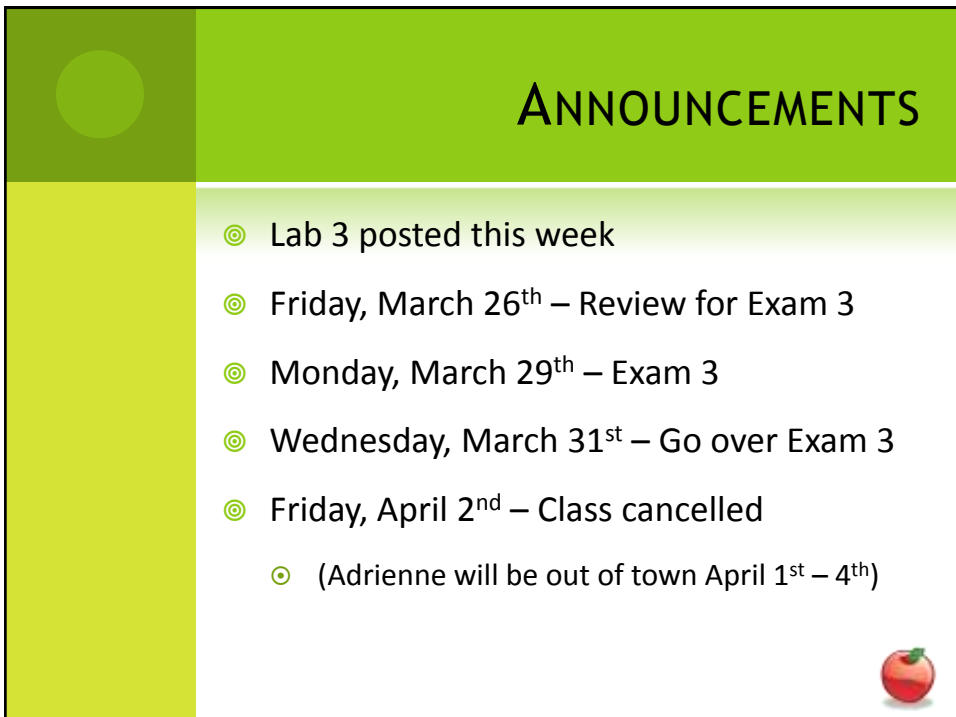



CSE 113 B

March 22-26, 2010



ANNOUNCEMENTS

- ⦿ Lab 3 posted this week
- ⦿ Friday, March 26th – Review for Exam 3
- ⦿ Monday, March 29th – Exam 3
- ⦿ Wednesday, March 31st – Go over Exam 3
- ⦿ Friday, April 2nd – Class cancelled
 - ⦿ (Adrienne will be out of town April 1st – 4th)



3

CHAPTER 7 - DRAWING STARS

- ⦿ Drawing stars on the screen
 - ⦿ Create method for drawing stars and call it from constructor of Space
 - ⦿ Inside method we retrieve the background image and draw ovals at random locations
 - ⦿ We also added functionality to create stars in random shades of gray.



4

SPEEDING UP ROCKET

- ⦿ Create code so that the rocket will show a different image when the user selects to speed it up.



5

CASTING

- ◎ Recall from earlier examples the following code:

```
Actor a = getOneIntersectingObject(X.class);
```

- ◎ Remember that X is the class we are interested in looking for collisions with – it can be anything (Flower, Ball, Brick, Barrel).
- ◎ `getOneIntersectingObject` returns the object we are interested with or null if not intersecting an object of the passed-in type. The object that is passed back is of type Actor.



6

CASTING

- ◎ Therefore, the type of the variable a is Actor.
- ◎ If we try to do this:

```
X a = getOneIntersectingObject(X.class);
```

- The code will not compile because `getOneIntersectingObject` returns an Actor, not an X.
- But we know that the Actor that is really being returned is an X.



7

CASTING

- ⦿ However, sometimes we may want to do things with a (the variable) that only X's can do.
- ⦿ However, a is an Actor and can only do things Actors can do.
- ⦿ If we want to treat the object that is returned by `getOneIntersectingObject` as an X, we can explicitly cast it as an X.



8

CASTING

```
X a = (X) getOneIntersectingObject(X.class);
```

- ⦿ The (X) is the cast.



9

PROTON WAVE ANIMATION

- ⦿ In ProtonWave class, we see a number of new things:
 - ⦿ Array
 - ⦿ While loop
- ⦿ Each of these things is explained in greater detail in Chapter 5. We are not covering the example from Chapter 5, but these concepts are being covered.



10

ARRAYS

- ⦿ Another type of collection (way to keep track of a group of objects).
- ⦿ Arrays are fixed size.
- ⦿ To declare a variable that holds an array:
`TypeOfThingInArray[] name;`
- ⦿ To create an array and assign it to the variable:
`name = new TypeOfThingInArray[NUMBER];`
 - ⦿ Where number is the number of elements you can store in the array.



11

ARRAYS

- ⦿ You can access elements in a array by using their index.
- ⦿ Indices for an array are from 0 to size -1. So, if there are 20 elements in an array, valid indices are 0-19.

```
nameOfArray[index]
```

- ⦿ Would allow you to access the element at that index

```
nameOfArray[index] = blah;
```

- ⦿ Would assign blah to that index.



12

WHILE-LOOP

- ⦿ Another form of iteration (looping).
- ⦿ This loop is not a counting loop like the for-loop, but rather will keep looping until the condition indicated on the loop is false.

```
while (booleanExpression)
{
    //code that should be repeated
}
```

