CSE 113 B

September 27 – October 1, 2010

Announcements

- Exams Returned Monday 9/27
- Lab 4 posted

3

Exam 1 Stats

Exam Statistics	
Min	52
Median	79
Average	78.84
Max	100
Std Dev	10.49

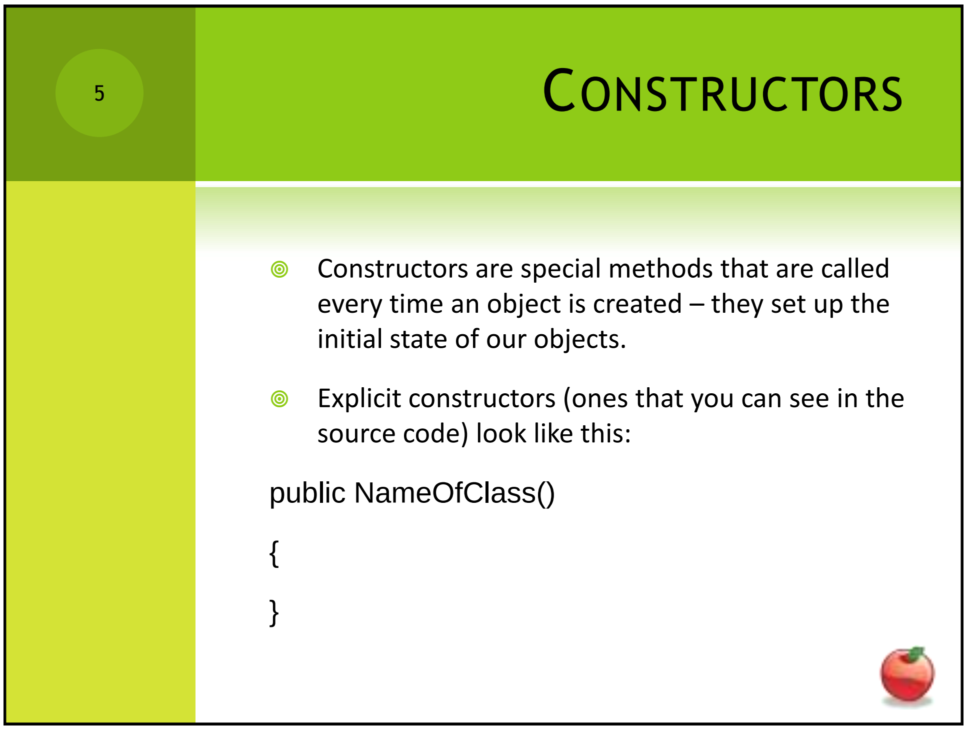
Grade Breakdown	
A	21%
A-	5%
B+	22%
B	22%
B-	12%
C+	8%
C	4%
C-	3%
D	3%
F	0%

4

Last Year's Stats

Exam 1 Grades SP 2010	
A	22%
A-	14%
B+	14%
B	15%
B-	10%
C+	6%
C	6%
C-	3%
D	4%
F	6%

Overall Letter Grades - SP 2010	
A	14%
A-	14%
B+	10%
B	12%
B-	9%
C+	7%
C	6%
C-	2%
D	3%
F	11%
R	12%

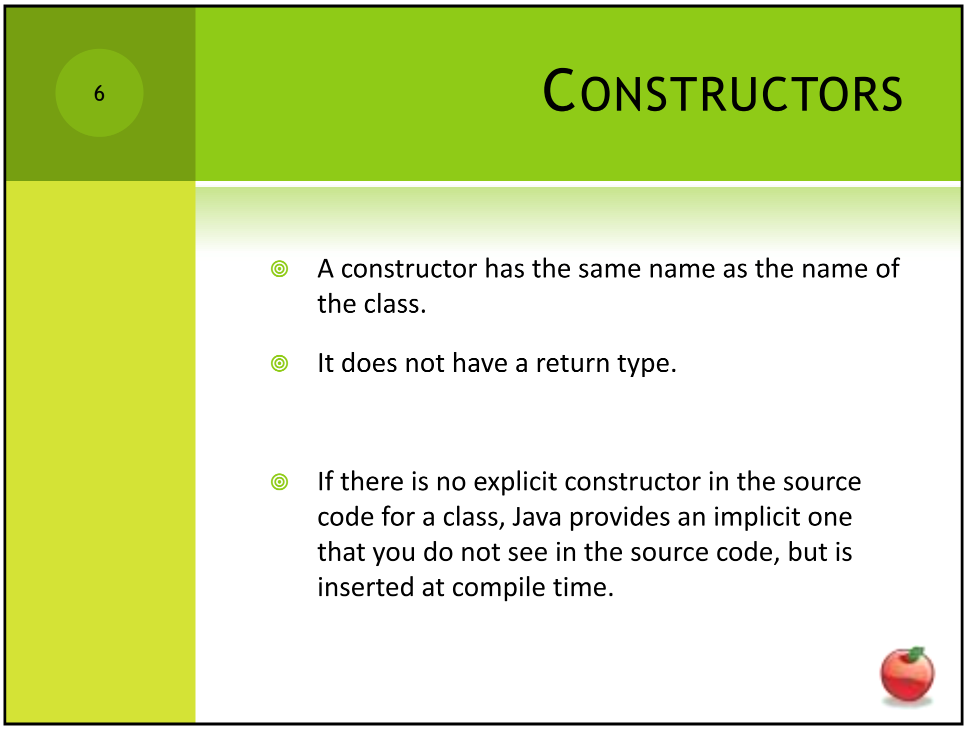


5

Constructors

- Constructors are special methods that are called every time an object is created – they set up the initial state of our objects.
- Explicit constructors (ones that you can see in the source code) look like this:

```
public NameOfClass()
{
}
```



6

Constructors

- A constructor has the same name as the name of the class.
- It does not have a return type.

- If there is no explicit constructor in the source code for a class, Java provides an implicit one that you do not see in the source code, but is inserted at compile time.

7

CarWorld Class

- Looking at the constructor of CarWorld, we can see a method call that looks like this:

```
super(x, y, z)
```

- Here, we are not calling a method called super, but rather super is a keyword that indicates the superclass. In this case, we are calling the superclass' constructor.

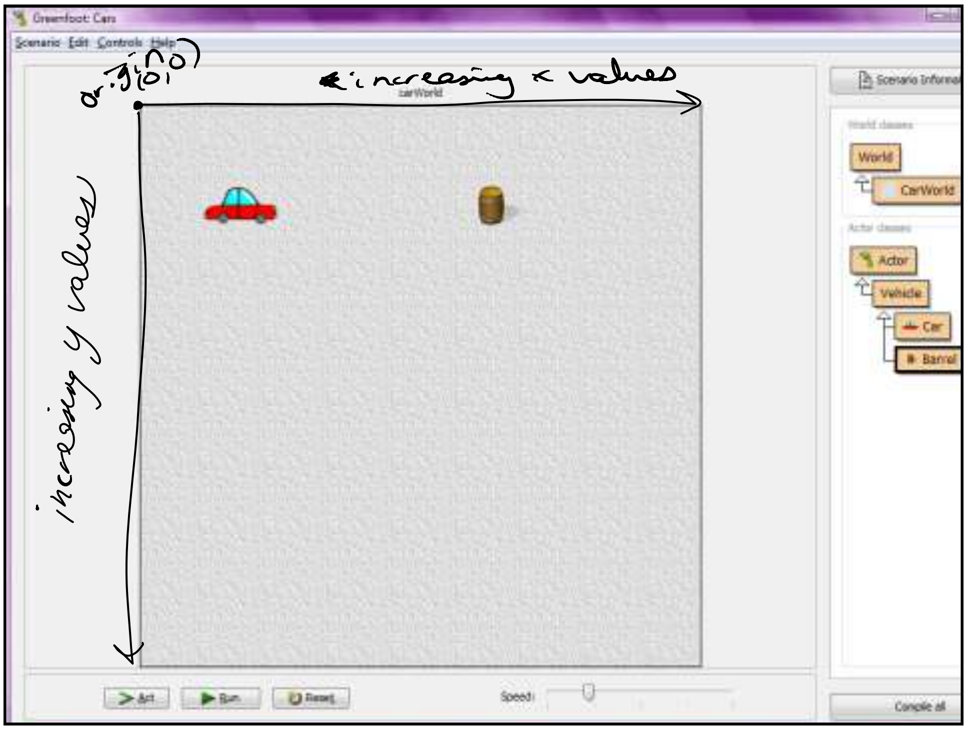
8

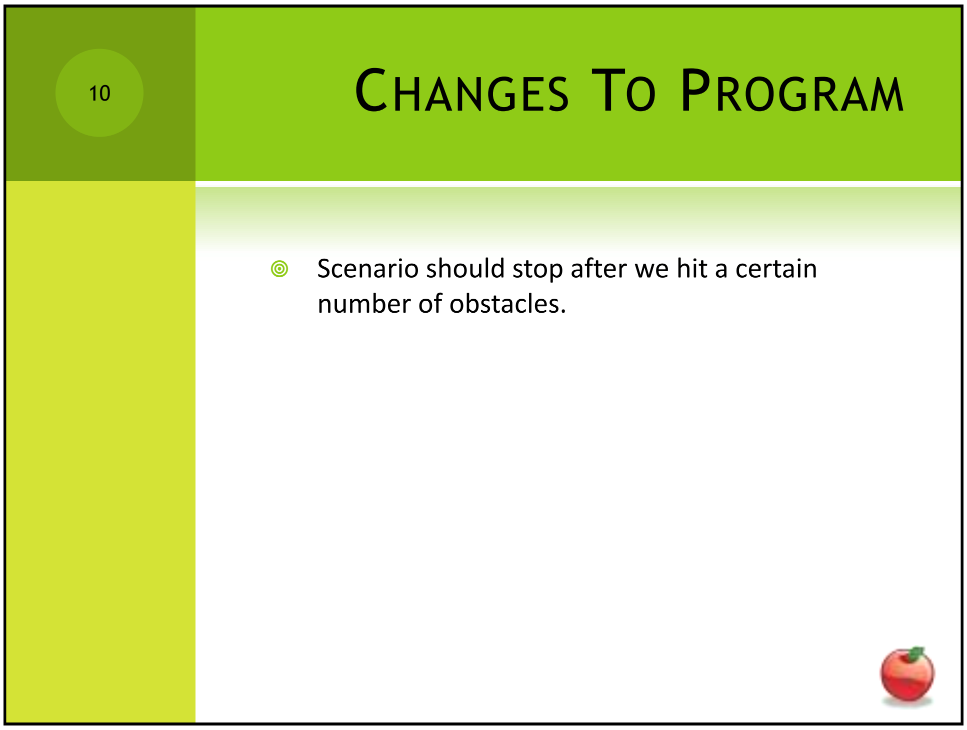
Adding objects at startup

- We can add objects to the world when it is created by calling the addObject method from the world.
- Example

```
addObject(new Car(), 34, 56);
```

- Note that we need to create a new Car object to add by using the expression new Car(). This expression creates an object and calls the constructor of that object.
- The numbers that follow are the x and y coordinates of where we would like the object to be in the world.





10

Changes To Program

- Scenario should stop after we hit a certain number of obstacles.

11

Question

- What do we need to do?

12

Answer

- First, recognize that we've hit something:

```
if(hit barrel)
{
    //Q: What do we do?
}
```

1- Turning away or removing barrel

2- Keeping track of how many we've hit

3- Notice if we've gone over the limit

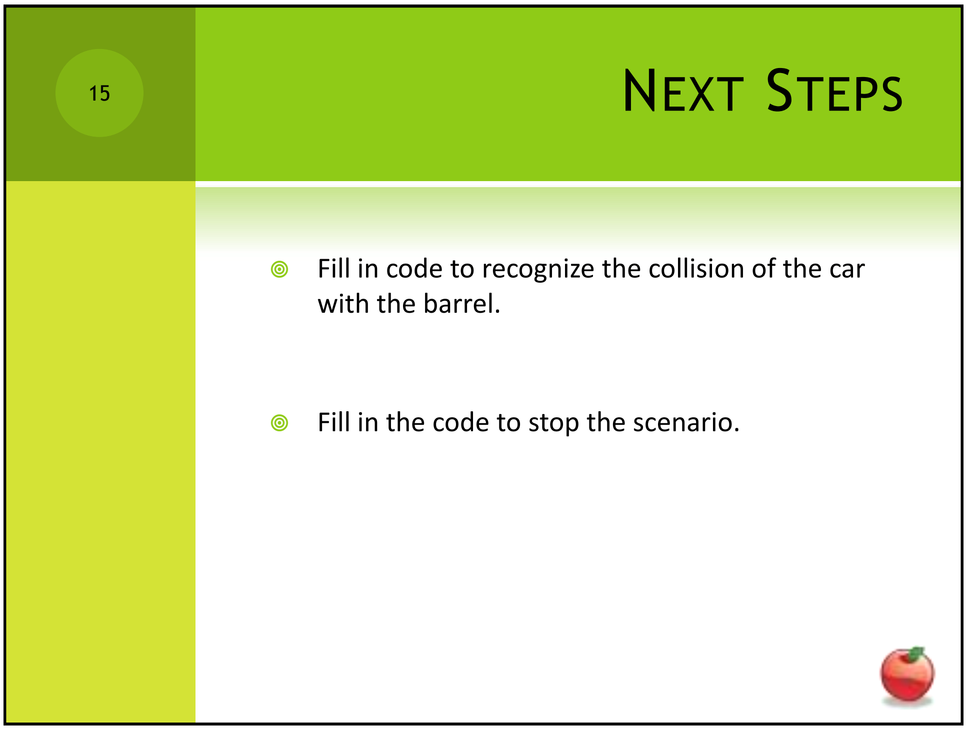
13

```
if(hit barrel)
{
   turn away
   note that we hit another barrel
   if(we've hit too many)
   {
      stop scenario
   }
}
```

14

Code that will run (but not perfectly)

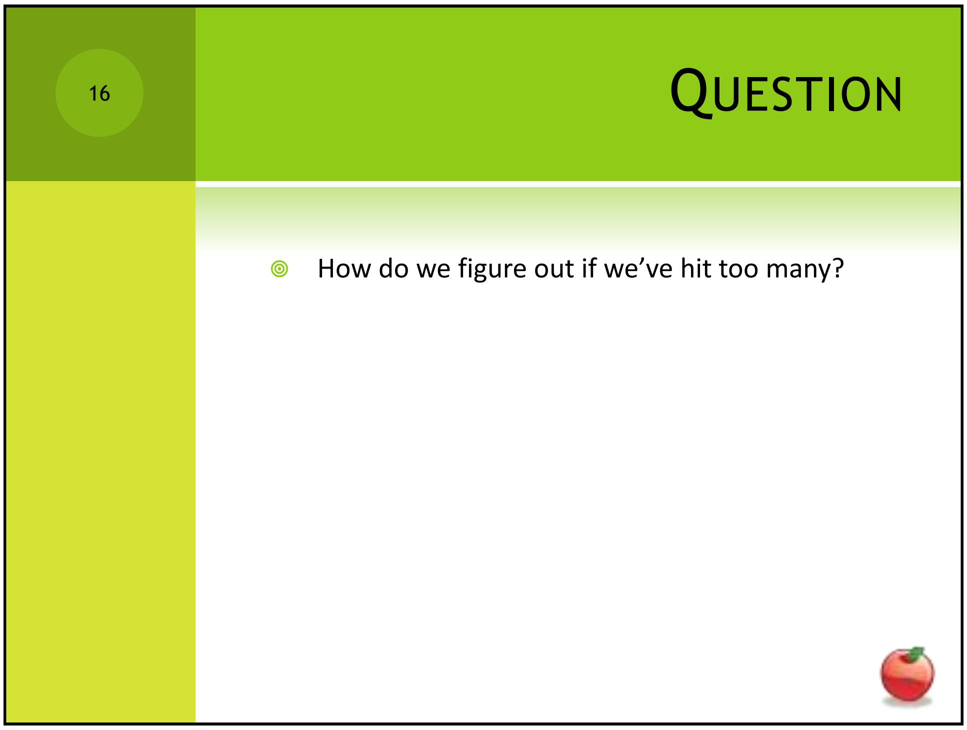
```
if(true) //hit barrel
{
   turn(45);
   //note that we hit another barrel
   if(false) //we've hit too many
   {
      //stop scenario
   }
}
```



15

Next Steps

- Fill in code to recognize the collision of the car with the barrel.
- Fill in the code to stop the scenario.



16

Question

- How do we figure out if we’ve hit too many?

17

VARIABLES

- Variables are used to store information.
- Instance variables store information important to the entire class.

18

INSTANCE VARIABLE SYNTAX

variable at the class level

private Type identifier ;

Type of information the variable stored

name for variable (programmer picks)

19

VARIABLES

- After we declare the instance variables, it is good practice to give it an initial value.
- We would give an instance variable and initial value in the constructor of the class.
- Example

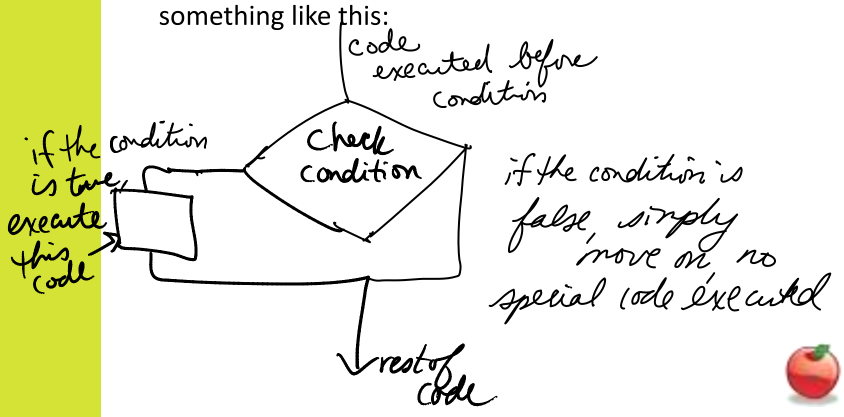
```
_barrelsHit = 0;
```

- Note that this expression uses the assignment operator (=) and takes the values on the right hand side and assigns them to the variable on the left hand side.

20

IF-STATEMENTS

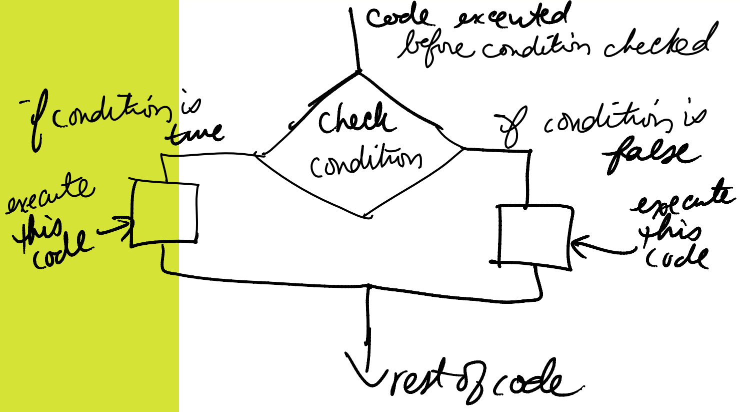
- We have been working a lot with if-statements to determine choices in our programs. If we look at our program execution with if-statements, it would look something like this:


code executed before condition
Check condition
if the condition is true, execute this code
if the condition is false, simply move on, no special code executed
rest of code

21

More ways to choose

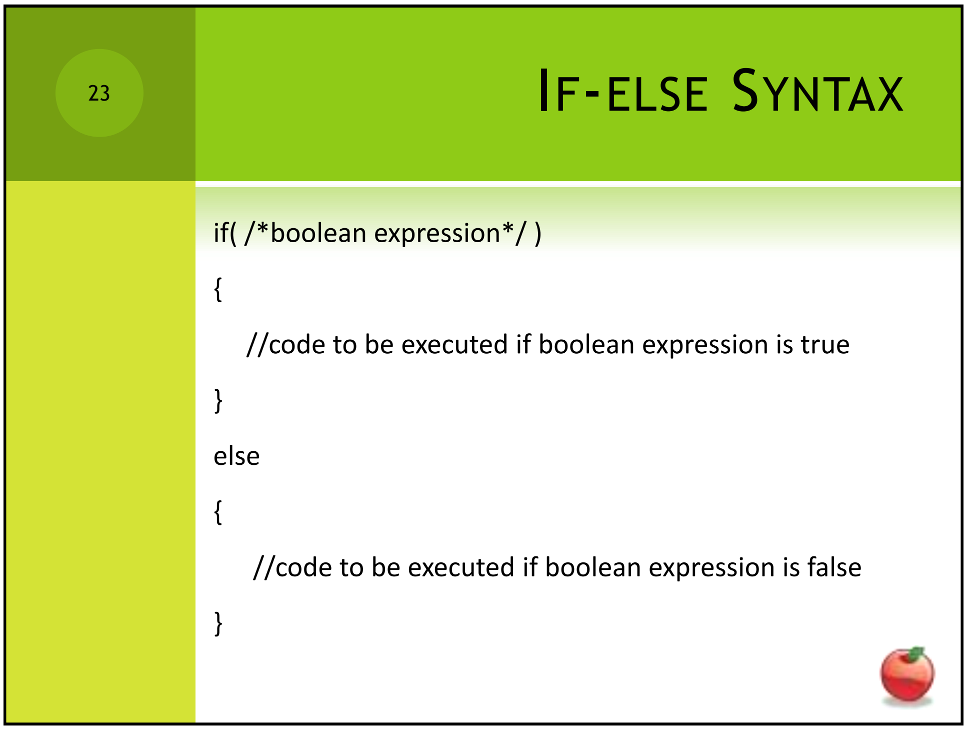
- We could create choice in programs that looks like this:



22

More ways to choose

- That would be the notion of a choice when there is a definitive path when a condition is true and another path when the condition is false.
- In order to do this type of choice in code, we would need to use if-else statements instead of just if-statements.



23

IF-ELSE SYNTAX

```
if( /*boolean expression*/ )
{
    //code to be executed if boolean expression is true
}
else
{
     //code to be executed if boolean expression is false
}
```