Compound Booleans

Mr. Neat

Java

Compound Booleans

What if you wrote a program that asked “Who was buried in General Grant’s Tomb?”

User’s could respond:

-Grant

-GRANT

-grant……all are correct.

Using what you know...

Grant

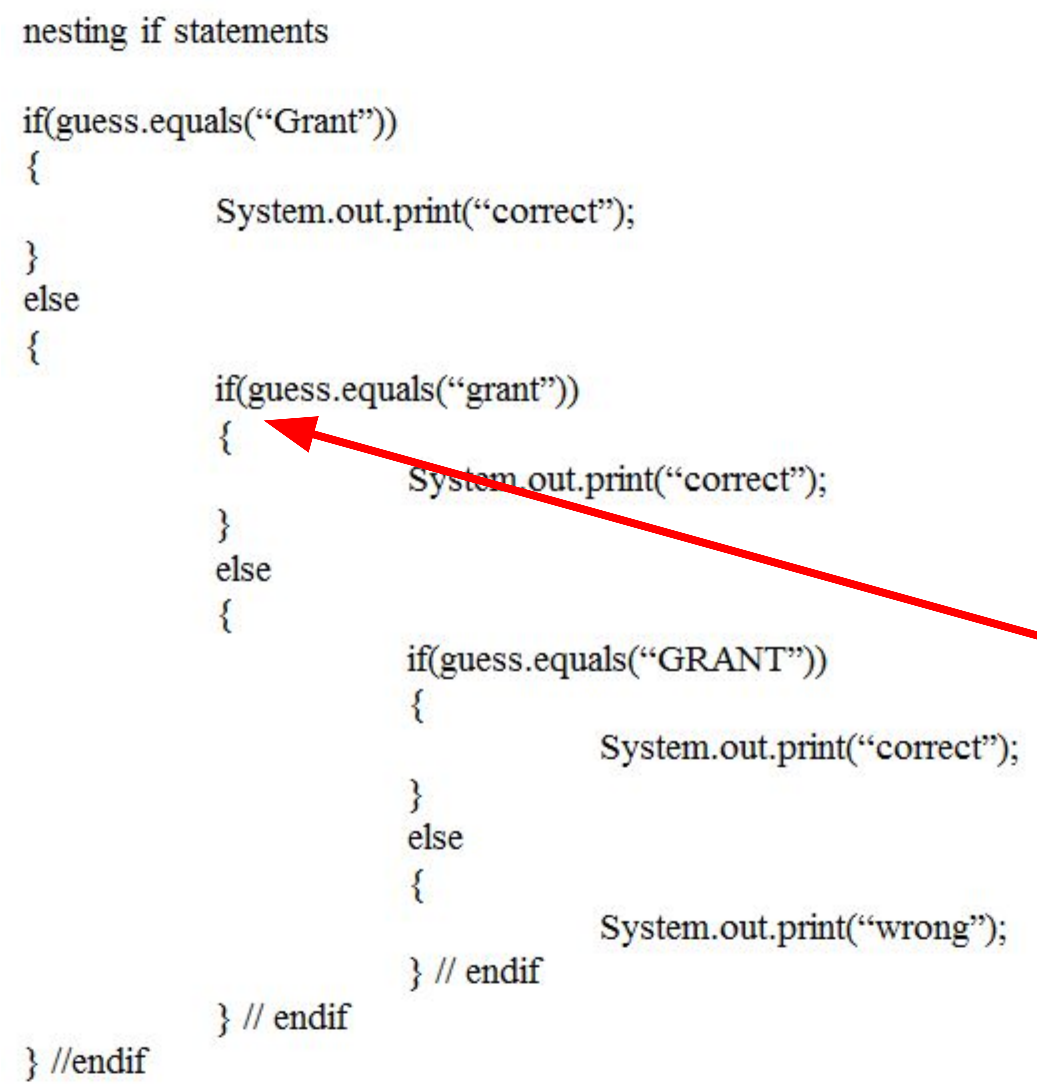
grant

GRANT

nesting if statements

```
if(guess.equals("Grant"))
{
        System.out.print("correct");
}
else
{
        if(guess.equals("grant"))
        {
                System.out.print("correct");
        }
        else
        {
                if(guess.equals("GRANT"))
                {
                        System.out.print("correct");
                }
                else
                {
                        System.out.print("wrong");
                } // endif
        } // endif
} //endif
```

Using what you know...



nesting if statements

```
if(guess.equals("Grant"))
{
        System.out.print("correct");
}
else
{
        if(guess.equals("grant"))
        {
                System.out.print("correct");
        }
        else
        {
                if(guess.equals("GRANT"))
                {
                        System.out.print("correct");
                }
                else
                {
                        System.out.print("wrong");
                } // endif
        } // endif
} //endif
```

This is called "nesting" if statements

else if statements

- Nesting can be in *if* part or *else* part.
- Nesting can be avoided with the use of *else if(condition)*.

Something new:
else if statements

```
if(boolean)
{

}
else if(boolean)
{

}
else if(boolean)
{

}
else
{

}
```

only one of these will fire

Something new: Compound Booleans

```
if(guess.equals(“grant”)||guess.equals(“Grant”))
{
    System.out.println(“correct”);
}
```

first boolean

|| means OR

second boolean

Something new: Compound Booleans

if(guess.equals(“grant”)||guess.equals(“Grant”))

- can have as many booleans as you want
- only one has to be true for the whole boolean to be true
- Java stops evaluating the booleans as soon as it finds a true boolean (short circuiting)

Something new:
Compound Booleans

Also have an “and” boolean....&&

```
int ex = 55;
int why = 17;

if(ex > 100 && why < 200)
{
    System.out.println(“pizza”);
}
```

Something new: Compound Booleans

```
if(ex > 100 && why < 200)
```

- can have as many booleans as you want
- all booleans have to be true for the whole boolean to be true
- Java stops evaluating the booleans as soon as it finds one false boolean (short circuiting)

Lab

- Add another drive method to your Car class
- The header is:
  - public void driveRandom()
- This method:
  - moves the Car one step
  - recycles the Car when the Car goes off the screen to the right
  - if the Car goes off the top or bottom of the screen, the y recycle location is the upper left corner of the screen
  - if the y location at the time of recycling is anywhere else (any road) then the y recycle location should be a random choice between:
    - same road it was just on
    - the road above it
    - the road below it
- Test it with 2 Car objects and SOP their y locations