

WES-CS GROUP MEETING #5

Exercise 1: Boolean conditions

Play a game using the decks of yellow and green cards. Each green card has an English sentence that is either true or false. Each yellow card has one of three symbols: !, &&, or || (meaning *not*, *and*, *or*). The game is played as follows:

- Each player starts with 6 cards, 3 yellow and 3 green.
- When it's your turn, if you can make a logical formula that evaluates to true using at least 4 of your cards, put down the formula and draw new cards of the same colors as the ones you used. Otherwise, trade in two of your cards for two new cards of the same color.
- The first person to put down at least 13 cards wins.
- You can use parenthesis cards whenever you want, but they don't count toward the 4 cards you need to use, or toward the 13 cards you need to win.

The precedences of the logical operators are as follows:

!	highest precedence
&&	next highest precedence
	lowest precedence

Exercise 2: Pounds and Pence (Algorithms & Programming)

In England, instead of dollars and cents, people use pounds and pence. There are 100 pence to a pound, and people say “p” instead of pence (for example, “twenty p” instead of “twenty pence”). The following coins are currently in use: 1p, 2p, 5p, 10p, 20p, 50p, 1 pound, and 2 pounds.

For this exercise, you will figure out how to give someone a specified amount of money (between 1p and 5 pounds) using the smallest possible number of coins. For example, if the amount is 69p, you would give these coins: one 50p, one 10p, one 5p, and two 2p.

Part (a).

Work with one or two others to design an algorithm that solves this problem. Write down your algorithm and trade algorithms with another group. Your team leader will give you some inputs to use with the algorithms. Execute the algorithm you have step by step, while the authors watch. If you think the other group is making a mistake with your algorithm, talk to them about it. If this exercise uncovers an error in your algorithm, fix it.

Part (b)

Now divide into three groups each using one of the laptops. If different people in your group have different algorithms, choose one. Write the `makeChange` method of the `ChangeMaker` class to implement the algorithm that you chose.

- The method will have one `int` parameter, `pence`, the amount to be converted to change. For example, if the amount to be converted is 1 pound and 4 pence, the `pence` parameter will be 104.
- The method will determine how many of each coin to use.
- It will display the coins to be used in order from the largest coin (2 pounds) to the smallest (1 p).

Test your code on a number of inputs. When everyone is happy with their code, compare to see if your solutions are the same or different.

For Discussion:

- If different people came up with different algorithms or different code, what were the advantages and disadvantages of the different versions?
- Do you think it was a good idea to write down your algorithms and to try them out before actually writing Java code? Why or why not?

Exercise 3: If I only had a Rectangle... (Java Coding)

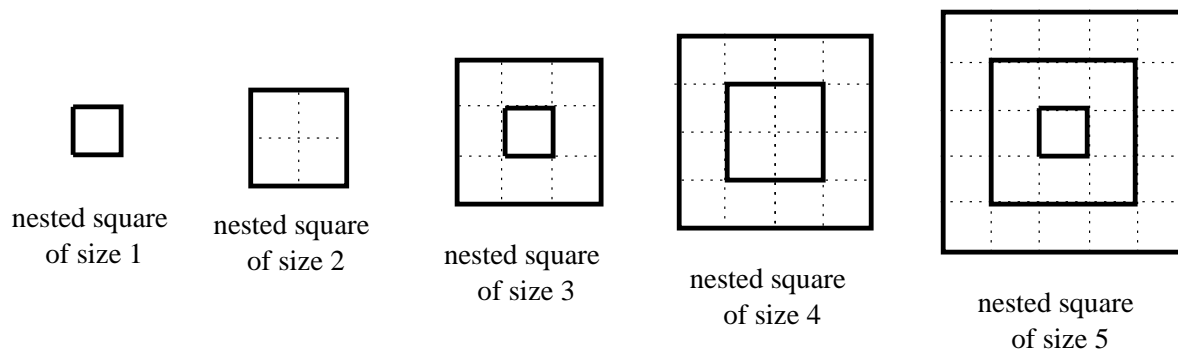
During week two we learned about the `Artist` class that we could use to draw simple pictures using method calls. Many of you discovered that some pictures would have been much easier to program if the `Artist` class had a method to draw a rectangle.

Part (a).

Divide into three groups, each using one of the laptops. Add a method to the `Artist` class that draws rectangles using the methods that were originally provided (no loops needed!). Be sure to check that your method works as you expect.

Part (b).

The pictures below show four *nested squares* (the dotted lines are just there to show you the grid of cells).



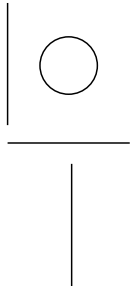
Add a method to the `Artist` class that draws nested squares. The method should have one `int` parameter that tells you the size of the outermost square.

Part (c).

What changes would you have to make to write a method that draws *nested rectangles*?

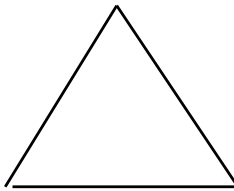
Exercise 4: Thinking Outside the Glass (Logical Thinking)

Part (a): Use 4 toothpicks and a Hershey's kiss to make the following picture of a cocktail glass with an olive in it:



Problem: Move only two of the toothpicks to change the picture to a glass (the same shaped glass) with the olive *outside* the glass.

Part (b): Now create an equilateral triangle using three toothpicks:



Problem: Add three more equilateral triangles of the same size as the original using only three more toothpicks.

Green cards

The Earth is flat

Madison is the capital of Wisconsin

In Java, * has higher precedence than +

In Java, + has higher precedence than -

The moon is made of green cheese

17 is a prime number

There is not a DDR machine in Memorial Union

The dorms get Cartoon Network

Red and blue make purple

A chestnut tree makes acorns

Green cards

A giraffe has a black tongue

Ducks can't fly

We are learning C++ in CS302

To compare two Java strings, use `.equals` not `==`

The sun moves around the Earth

The UW-Madison was founded in 1492

There are 10 WES-CS groups

This room is CS 1240

Miss Muffet is afraid of spiders

Green cards

The time now is 3:33pm

The sun sets in the west

Java programs start executing in a class named Main

The following code will compile: `boolean b = "true";`

An if statement will repeat until the condition is false

Bascom Hall is exactly one mile from the state capitol building.

`System.out.println()` is a static method.

Booleans can have three possible values.

Boolean algebra was invented by Bobby Boolean.

Green cards

Maine is the only state whose name is just one syllable.

It is snowing outside.

The University of Wisconsin was founded in 1492.

Wisconsin became a state in 1911.

The mascot of the University of Wisconsin is Bucky the Bobcat.

Pascal, Python, and Scheme are all names of programming languages.

Yellow cards

&&

&&

&&

&&

&&

&&

&&

&&

&&

&&

&&

&&

Yellow cards

||

||

||

||

||

||

||

||

||

||

||

||

!

!

!

Yellow cards

!

!

!

!

!

!

!

!

!

Yellow cards

(

(

(

(

(

(

(

(

(

(

(

(

(

(

(