Before we start design and decisions ...

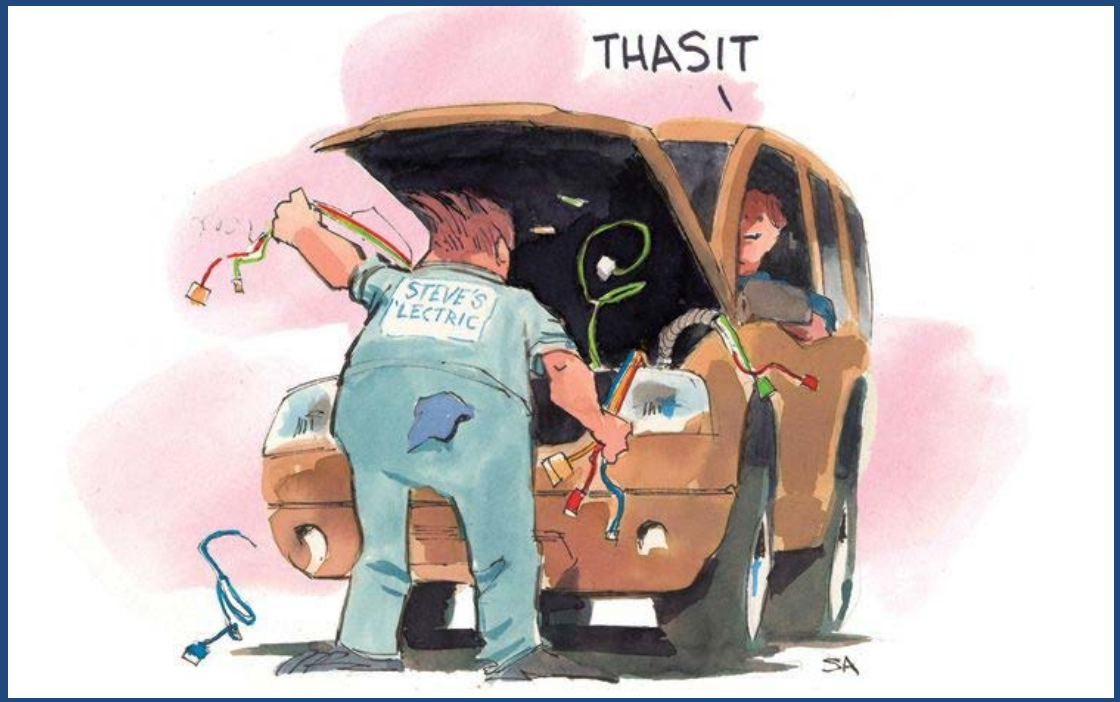
Let's do a quiz!

C Programming

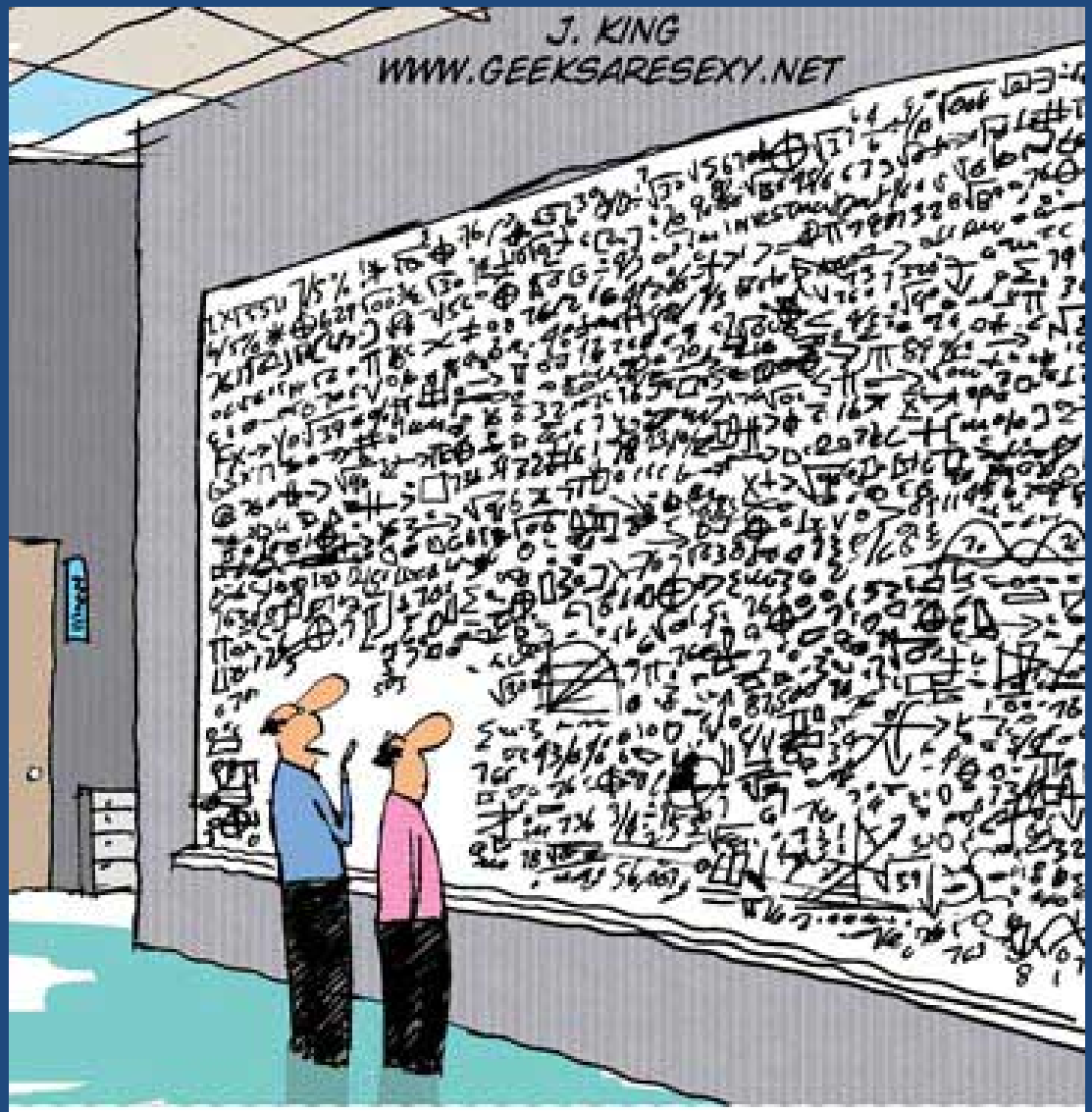
Design and Decisions

So far ...

mechanics!



You need
design skills too



"...And that, in simple terms, is what's wrong with your software design."

SEF will cover certain design topics in more detail later

You need
something to get you started
now

Pseudocode

It's an English representation of what the design should look like

English, not C

Use English-like statements to break down what you want to do

Start
at a
very high
level

For each English-like pseudocode statement, ask yourself this question ...

"Do I know how to do this easily?"

For each "no" answer, expand that one pseudocode line into more pseudocode

Review!

What's pseudocode used for?

Example

Problem:
"Describe how to make a peanut butter sandwich."

Think about what's done,

at a very high level

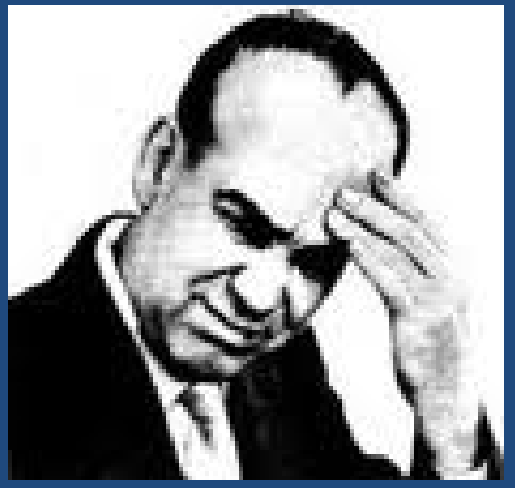
Get materials.

Prepare materials.

Make sandwich.

Ask yourself the question:

"Do I know how to do this easily?"



Get materials.

Do I know how to do this easily?

No.

I don't know what the materials are.

So, break it down.

Get materials.

Get bread.

Get peanut butter.

Get knife.

Get plate.

Then, for each statement we just made, ask the question:

Do I know how to do this easily?

I think I can answer "yes" to all four statements.

So I go on to the next statement.

Prepare materials.

Do I know how to do this easily?

No.

Open bread bag.

Get 2 slices of bread out of bag.

Put bread slices on plate.

Open peanut butter jar.

Do I know how to do these easily?

I'll say yes.

Go on to the next statement.

Make sandwich.

Do I know how to do this easily?

No.

Put peanut butter on one slice of bread.

Put other slice of bread on top of the first slice.

Cut bread.

Do I know how to put peanut butter on one slice of bread easily?

No.

Break it down.

Grab knife from handle with dominant hand.

Dip blade of knife into peanut butter.

Grasp bread on edges with other hand.

Get peanut butter on knife.

Spread peanut butter on bread evenly.

Repeat from second line aboveuntil enough peanut butter is on bread.

Do I know how to do each of those statements easily?

I'll say yes.

I'll also say that we know how to put the bread together

Cutting? No.

Grasp knife in dominant hand.
Cut diagonally across bread.
If four pieces desired, then
  Cut diagonally across bread in other direction.
End if

Why end if?

Indicate what is done in the case that the if statement is true

And so on.

One slight backtrack ...

In a previous breakdown, we repeated a series of actions

Replace that
with
a loop

Grab knife from handle with dominant hand.
Loop
    Dip blade of knife into peanut butter.
    Grasp bread on edges with other hand.
    Get peanut butter on knife.
    Spread peanut butter on bread evenly.
Repeat loop until enough peanut butter is on bread.

Combine it all,
using higher-level statements as comments

// Get materials

Get bread.

Get peanut butter.

Get knife.

Get plate.

// Prepare materials
Open bread bag.
Get 2 slices of bread out of bag.
Put bread slices on plate.
Open peanut butter jar.

// make sandwich

// put peanut butter on one slice of bread
Grab knife from handle with dominant hand.
Loop
    Dip blade of knife into peanut butter.
    Grasp bread on edges with other hand.
    Get peanut butter on knife.
    Spread peanut butter on bread evenly.
Repeat loop until enough peanut butter is on bread.

Put other slice of bread on top of the first slice.

```
// cut bread
Grasp knife in dominant hand.
Cut diagonally across bread.
If four pieces desired, then
   Cut diagonally across bread in other
   direction.
End if
```

Review!

What's the magic question?

Review!

What's do you do if the answer is yes?

Review!

What's do you do if the answer is no?

OK, now how about something that doesn't involve peanut butter

Problem: "Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

Not something that
a random person
would be able to code

But
that random person
could figure out how to do it

and they could do it with PSEUDOCODE!

Well, sort of

First, examine the problem

Look for
clues
in the
wording

"Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

We're dealing with a whole bunch of items

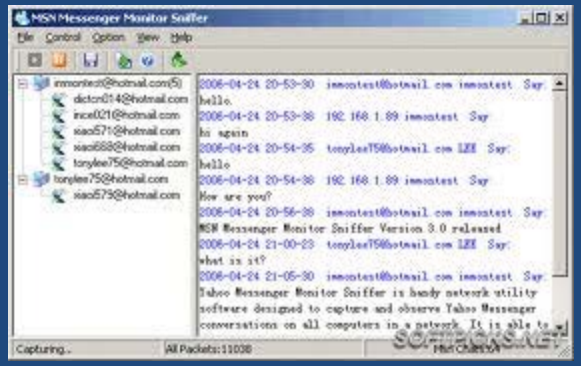
Loop!

"Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

Input!

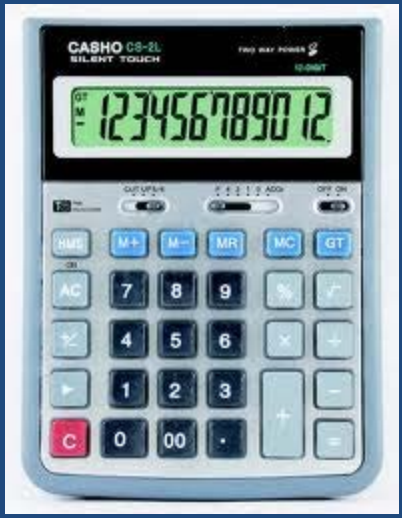
"Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

Output!


MSN Messenger Monitor Sniffer
File Control Option View Help
immontest@hotmail.com(5)
dictcn014@hotmail.com
ince021@hotmail.com
xiao571@hotmail.com
xiao668@hotmail.com
tonylee75@hotmail.com
tonylee75@hotmail.com
xiao579@hotmail.com
2006-04-24 20-53-30 immontest@hotmail.com immontest Say:
hello.
2006-04-24 20-53-38 192.168.1.89 immontest Say:
hi again
2006-04-24 20-54-35 tonylee75@hotmail.com LEE Say:
hello
2006-04-24 20-54-38 192.168.1.89 immontest Say:
How are you?
2006-04-24 20-56-38 immontest@hotmail.com immontest Say:
MSN Messenger Monitor Sniffer Version 3.0 released
2006-04-24 21-00-23 tonylee75@hotmail.com LEE Say:
what is it?
2006-04-24 21-05-30 immontest@hotmail.com immontest Say:
Yahoo Messenger Monitor Sniffer is handy network utility
software designed to capture and observe Yahoo Messenger
conversations on all computers in a network. It is able to
Capturing...
All Packets: 11038

"Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

Calculate!


CASHO
SILENT TOUCH
1234567890 12
HME
M+
M−
MR
MC
GT
AC
7
8
9
%
4
5
6
1
2
3
0
00
.
C

"Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

Decision!

"Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

At the end!

"Display the cost, including tax if applicable, for each of ten meals. Each meal's pre-tax price is entered by the user. When done, display the total cost for all meals."

Accumulation!

Sooooo, we know:

Loop for 10 meals
  Get input of pre-tax price for this meal
  Calculate total cost of this meal
  Display total cost of this meal
  Accumulate final total cost
End loop
Display final total cost

What would
your random person
likely say?

"Well, ask the person how much the meal is, then include the HST if it's above $4 or just include GST if it's not, and tell them. Keep doing that for the ten meals."

"Don't forget to keep track of how much the cost is so you can tell them a final figure at the end."

Loop for 10 meals
  Get input of pre-tax price for this meal
  Calculate total cost of this meal
  Display total cost of this meal
  Accumulate final total cost
End loop
Display final total cost

And, asking the magic pseudocode question ...

Review!

What's that question again?

Loop for 10 meals Yes

Get input of pre-tax price for this meal No

Calculate total cost of this meal No

Display total cost of this meal Yes

Accumulate final total cost No

End loop Yes

Display final total cost Yes

Get input of pre-tax price for this meal

Prompt user

Get pre-tax price from user

Calculate total cost of this meal

If cost of meal is above $4, then

  Multiply meal cost by 1.13

else

  Multiply meal cost by 1.05

Accumulate final total cost

final total cost = final total cost
+ cost for this meal

So it becomes ...

```
Loop for 10 meals
    // Get input of pre-tax price for this item
    Prompt user
    Get pre-tax price from user

    // Calculate total cost of this meal
    If cost of meal is above $4, then
            Multiply meal cost by 1.13
    else
            Multiply meal cost by 1.05
```

Display total cost of this meal

// Accumulate final total cost

final total cost = final total cost + cost for this meal

End loop

Display final total cost

Then convert it to C code

Review!

What's the relationship between pseudocode and C?

Review!

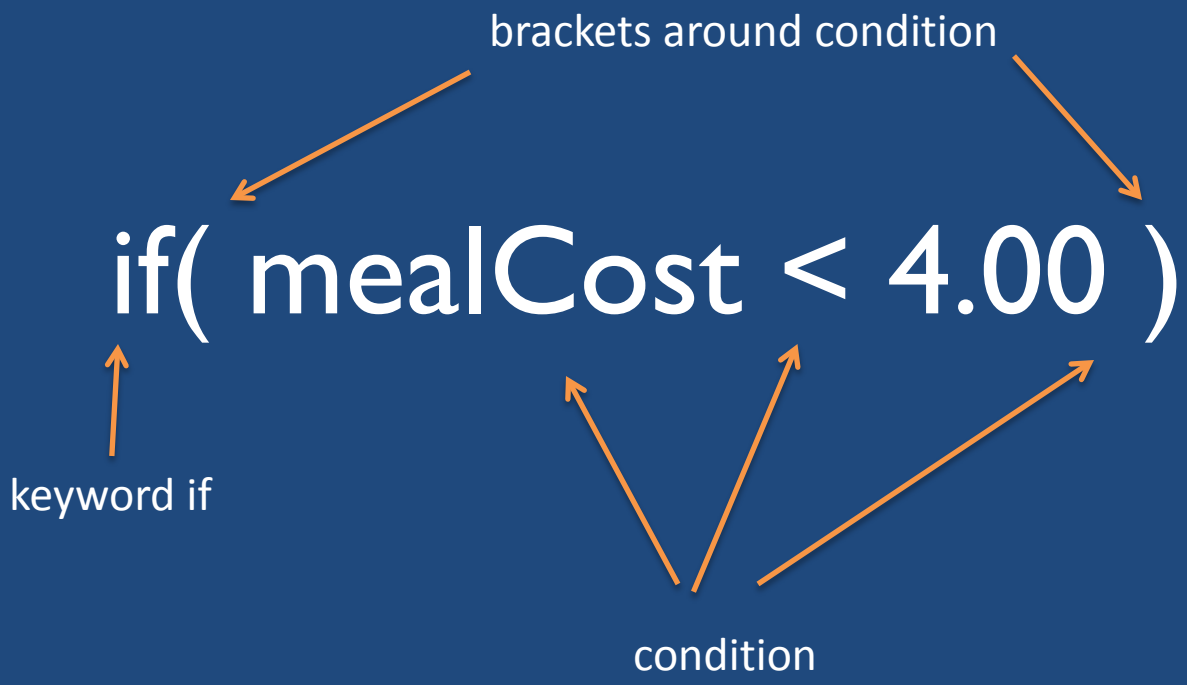
Why might you want to use pseudocode?

C decisions and loops?

Decisions that are not loops

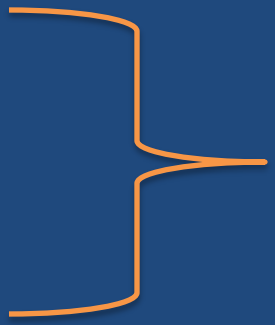
if statements

Example

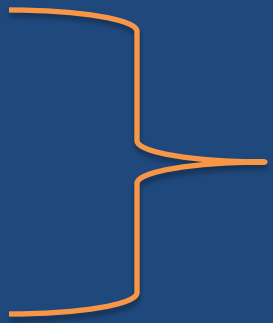

brackets around condition
if(mealCost < 4.00)
keyword if
condition

More complete example

```
if( mealCost > 4.00 )
{
   mealCost *= 1.13;
}
else
{
   mealCost *= 1.05;
}
```



body executed if true



body executed if false

The *else* is optional

The curly braces are optional in this case but ...

not in MY course!

Course Requirement

Put curly braces around single-line bodies.

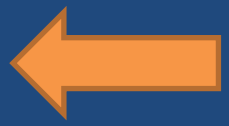
Review!

What's a body (in this context)?

Other Stuff

Don't put a semicolon after the if statement and before the body

```
if( mealCost < 4.00 ) ;
{
   mealCost *= 1.13;
}
else
{
   mealCost *= 1.05;
}
```



Bad!

You can't do a three-way comparison

Using

if(10 < a < 20)

won't work

You can chain if statements together by using else followed by if

```
if( counter < 100 )
{
      printf("The counter is %d\n", counter);
      counter += 4;
}
else  if ( counter < 200 )
{
      printf("The counter is getting pretty high\n");
}
else
{
      printf("The counter is too high\n");
}
printf("Counter is now %d\n", counter);
```

Not elseif

You can nest them too

```
if( counter < 100 )
{
   printf("counter is less than 100\n");
   if( counter < 0 )
   {
      printf("correcting a negative counter\n");
      counter = 0;
   }
}
```

Review!

What's an if statement used for?

Review!

What are the two different possible outcomes from an if statement?

How about looping?

Decisions that are loops

while statements

Aside: Yes, there are other ways of making decisions in C

How to decide about decisions?

Look at the description for:

- repeated actions (e.g. user input)
- plurals
- "while"
- "until"
- "repeat"

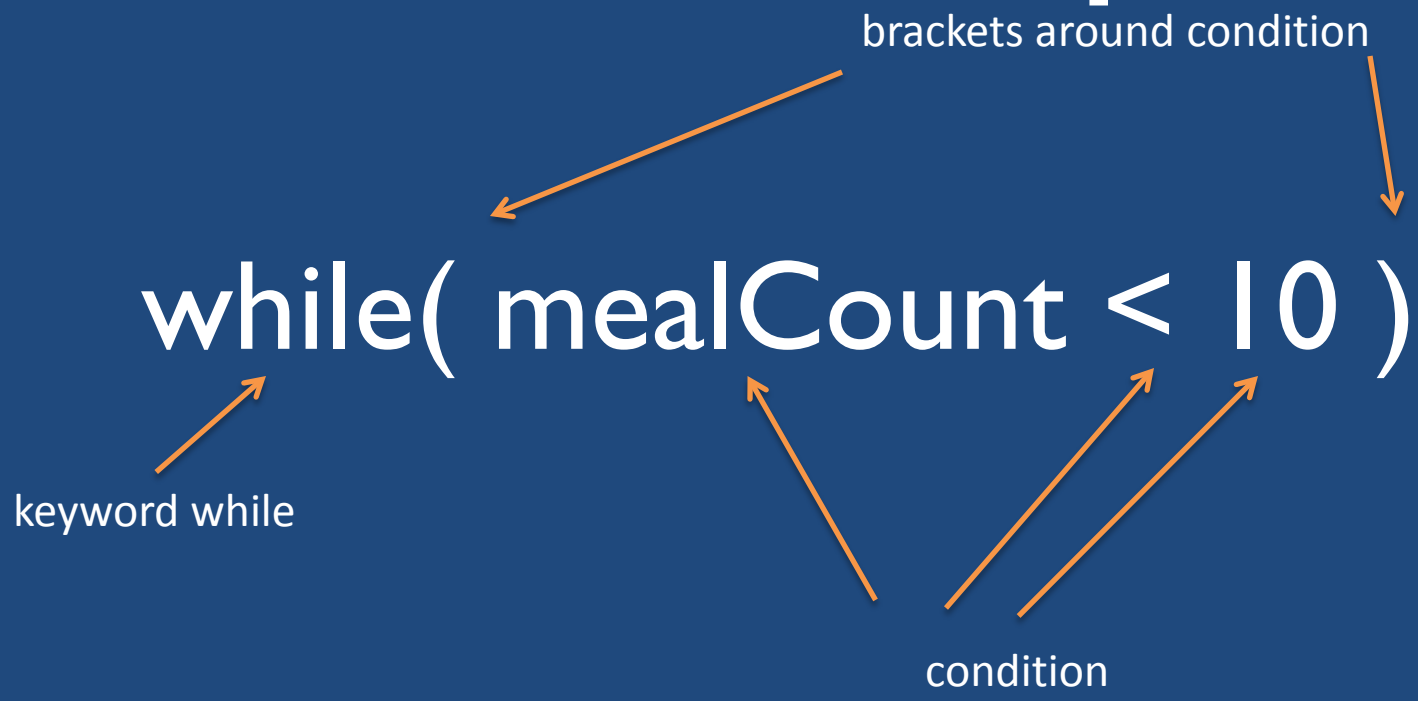
Loops!

Look at the description for:

- "if"
- "unless"
- "except"
- "case"
- other non-looping decision words

If!

while Example



Same syntax issues

as

if statements

Nesting while Loop Example

```
int i = 0, j = 0;
while(i < 10 )
{
    j = 0;
   while(j < 10 )
   {
       printf(" j is %d\n", j ++);
   }
   printf(" i is %d\n", i ++);
}
```

Review!

What should you use if you need to add together a whole bunch of numbers but you don't know how many?

Review!

What should you use if you need to calculate a shipping rate depending on if the recipient is in Canada or not?

Review!

What should you use if you need to calculate a shipping rate depending on if the recipient is in Canada, the U.S., or elsewhere?

Review!

What should you use to take different actions depending on what a user has entered?

Review!

What should you use to take different actions depending on what a user enters, ending when they simply press ENTER?

Logical Operators

So far, we've only seen

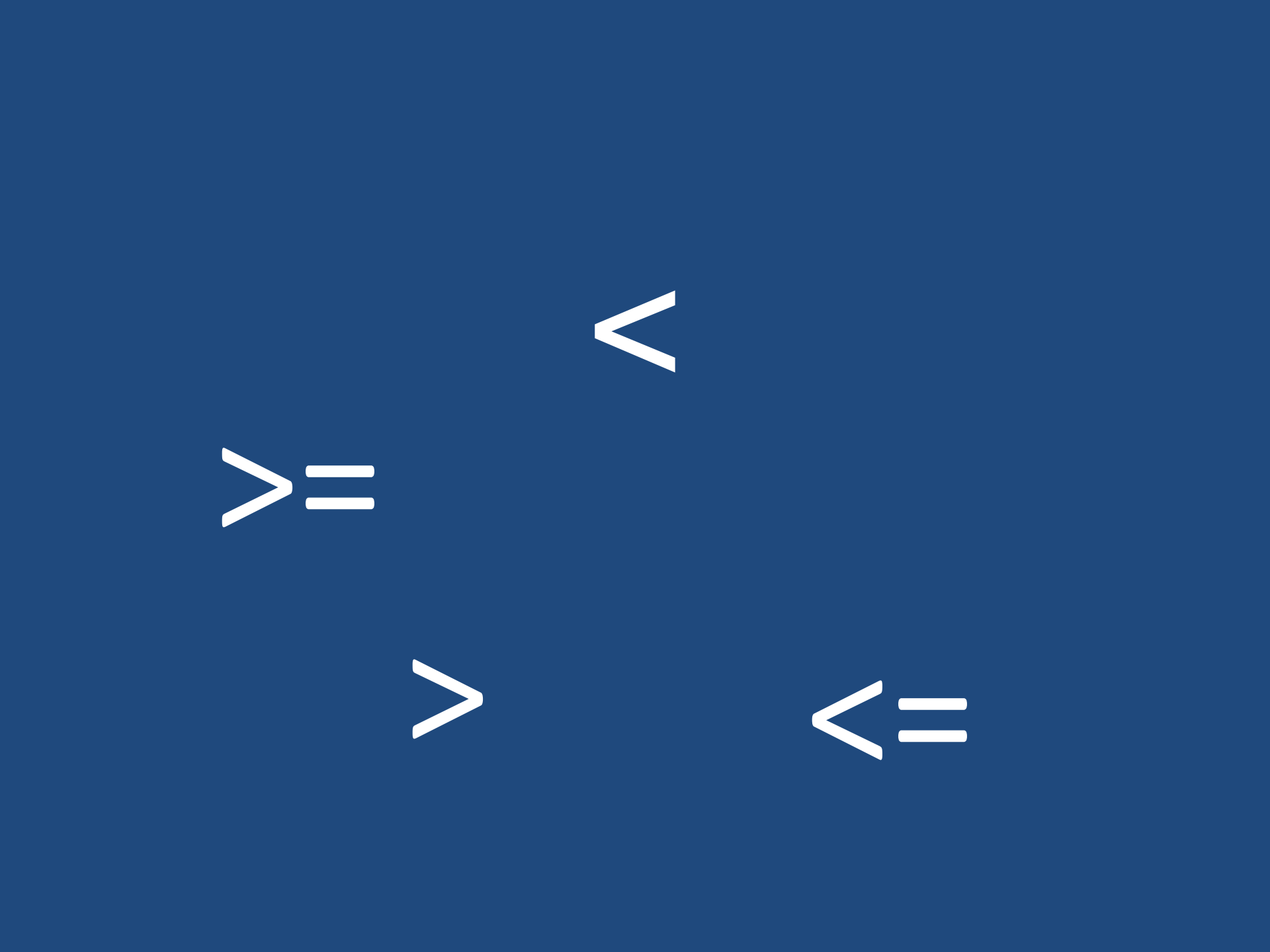
<

and

>

There are other comparison operators

(called logical operators)



Where's equals-to????

Yes, two of them

=

means

assignment

assignment:

```
a = b + c;
```

"Assignment"

"Change the variable"

if(a == 10)

is different from

if(a = 10)

if(a = 10)

means

"if the changing of the value of a to 10 will cause a to be a non-zero value (and, since 10 is non-zero, it will), then the condition is true"

Valid

but

bad!

A good habit to get into to avoid some of those:

backwards comparison!

Replace:

```
if( a == 10 )
```

with:

```
if( 10 == a)
```

so that forgetting will not be so bad:

```
if( 10 = a )
```

Compiler Error!

Review!

What does

if(a = 10)

mean?

Review!

What does

if(a == 10)

mean?

So, that's equality.

OK, how about Inequality?

Inequality is not <>

It's !=

Inequality example

```
if( a != 10 )
```

true and false

In C, false is equivalent to 0.

In C, true is equivalent to anything except 0.

bool

The *bool* datatype was introduced in VS2013

You can use it just like a normal data type, except ...

The only two possible values are

true

and

false

and

you must use

#include <stdbool.h>

More logical operators

You can also do logical negation:

```
if( ! (a == 10) )
```

And you can join conditions together using && and ||

Yes, that's a pair again

&&

is

logical-AND

||

is

logical-OR

&&

requires

both conditions to be true

for

the conjunction to be true

||

requires

either or both of the conditions to be true

for

the conjunction to be true

Example

if((value >= 5) && (value <= 9))

is looking for a value between 5 and 9, inclusive

Example

```
if( (value < 5) ||
(price > 10.5) )
```

is looking for a value less than 5
OR a price greater than 10.5
(or both)

A more complex example

if(((a == b) && (c == d)) || (e == f))

is a bit more difficult to parse but look at the extra brackets

It is a great idea to use more brackets than you have to to ensure the intended order of operations

Definitely review
the reading
if you are unsure
about any of this

Summary

1. You need to know how to design software.
2. You can use Pseudocode.
3. Decisions are important.
4. If statements and while statements are very useful.