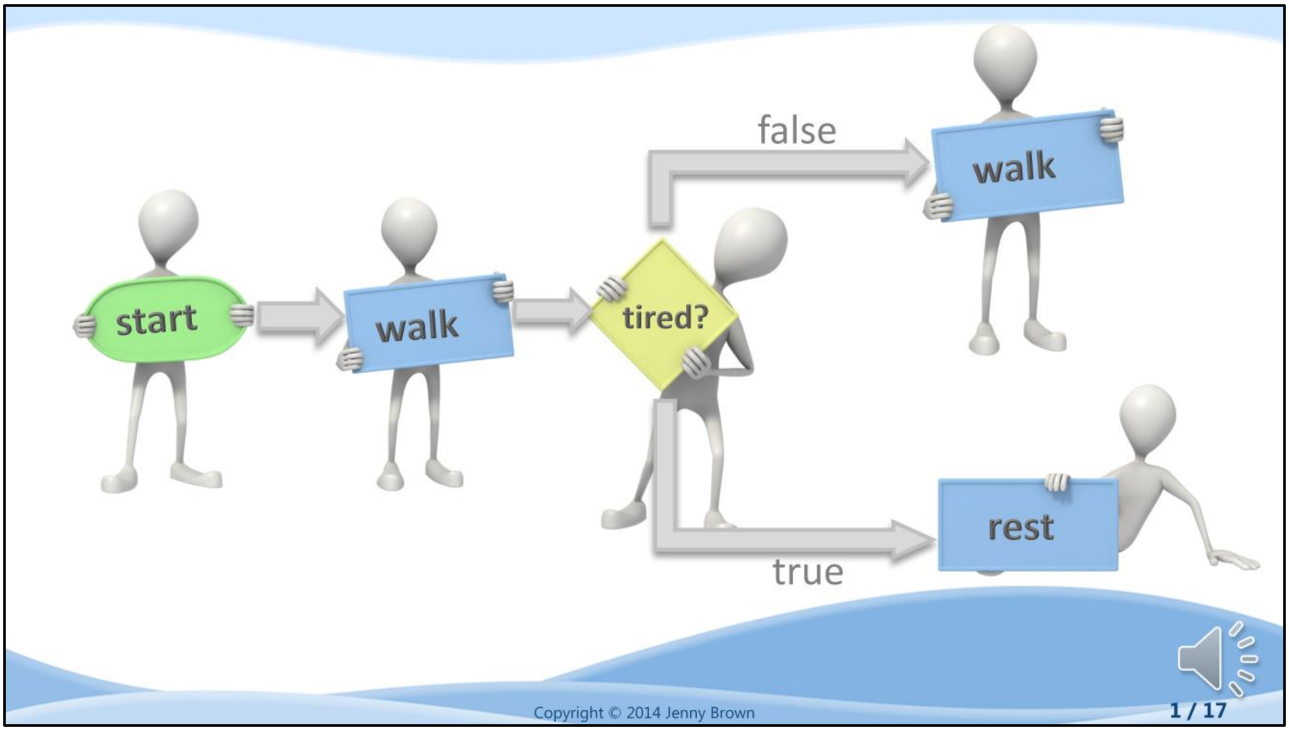
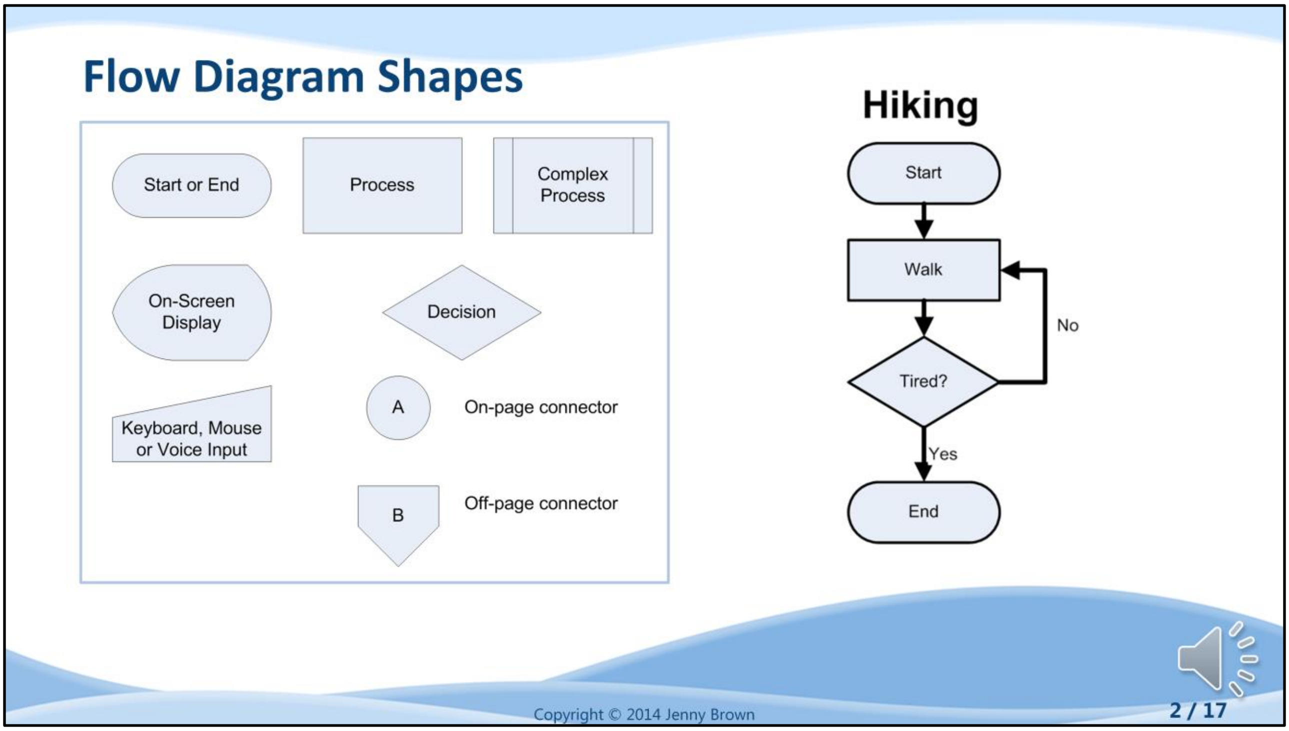


Copyright © 2014 Jenny Brown

1 / 17

Remember these guys? You've seen them at the beginning of each video, this whole time, with no explanation or context. Actually, they do have a good reason for being here. This video is going to discuss flowcharts: diagrams that help you plan out what software should do.

A software flowchart diagram lets us plan out the steps, carefully and in order, that a software program must go through to accomplish its task. Drawing a flow chart helps you plan your own software, and it can also help you solve practice and challenge problems in this course. It will stop you from getting confused as you tackle harder challenges.



Flow Diagram Shapes

Start or End

Process

Complex Process

On-Screen Display

Decision

Keyboard, Mouse or Voice Input

A On-page connector

B Off-page connector

Hiking

Start

Walk

Tired?

No

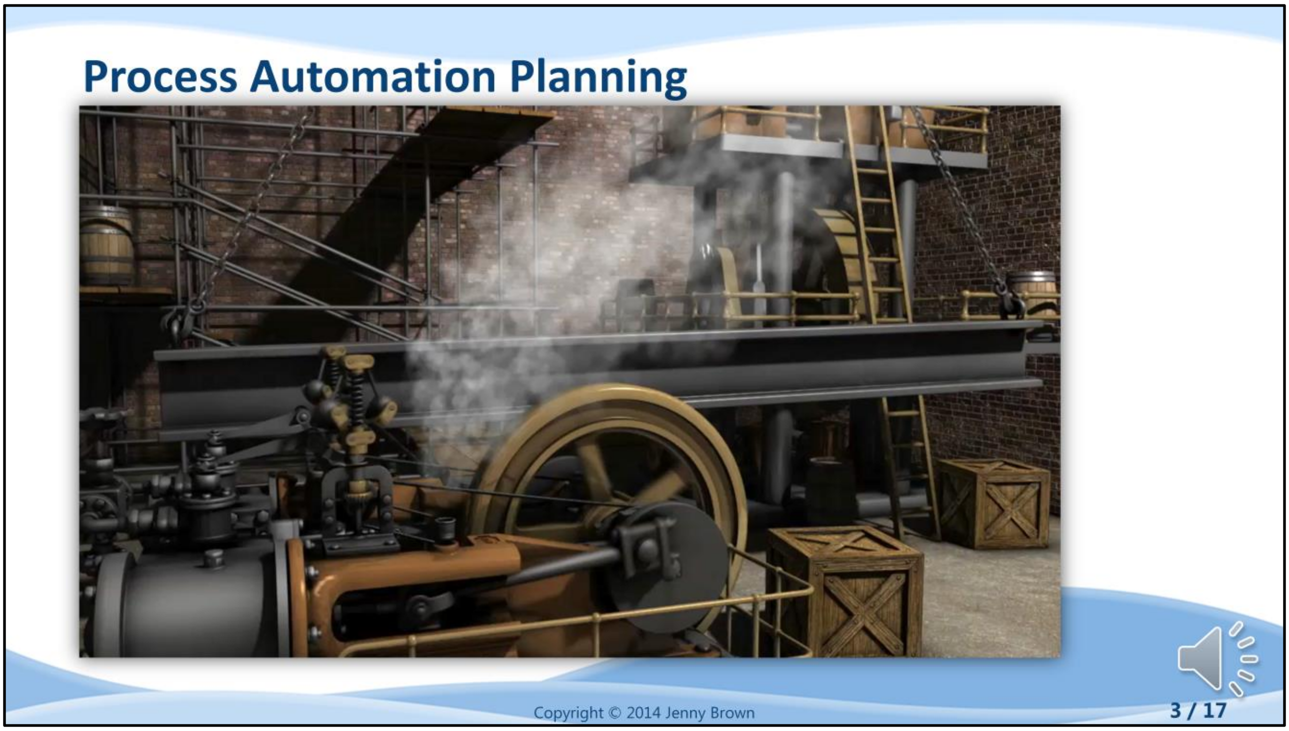
Yes

End

Copyright © 2014 Jenny Brown

2 / 17

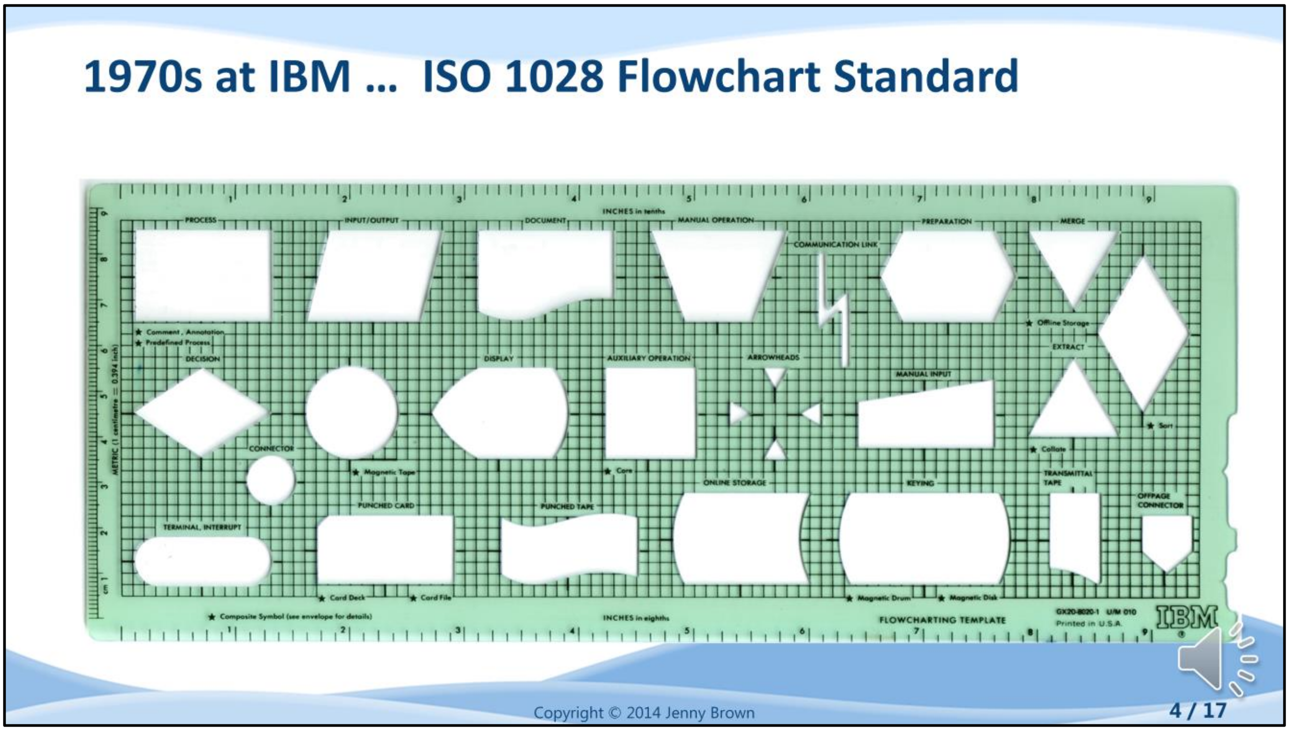
These shapes, plus hand-drawn arrows, will be used to plan the software we're working on in this course. You may have seen flow charts like this in internet jokes or other places. But where did flow charts come from?



Process Automation Planning

3 / 17

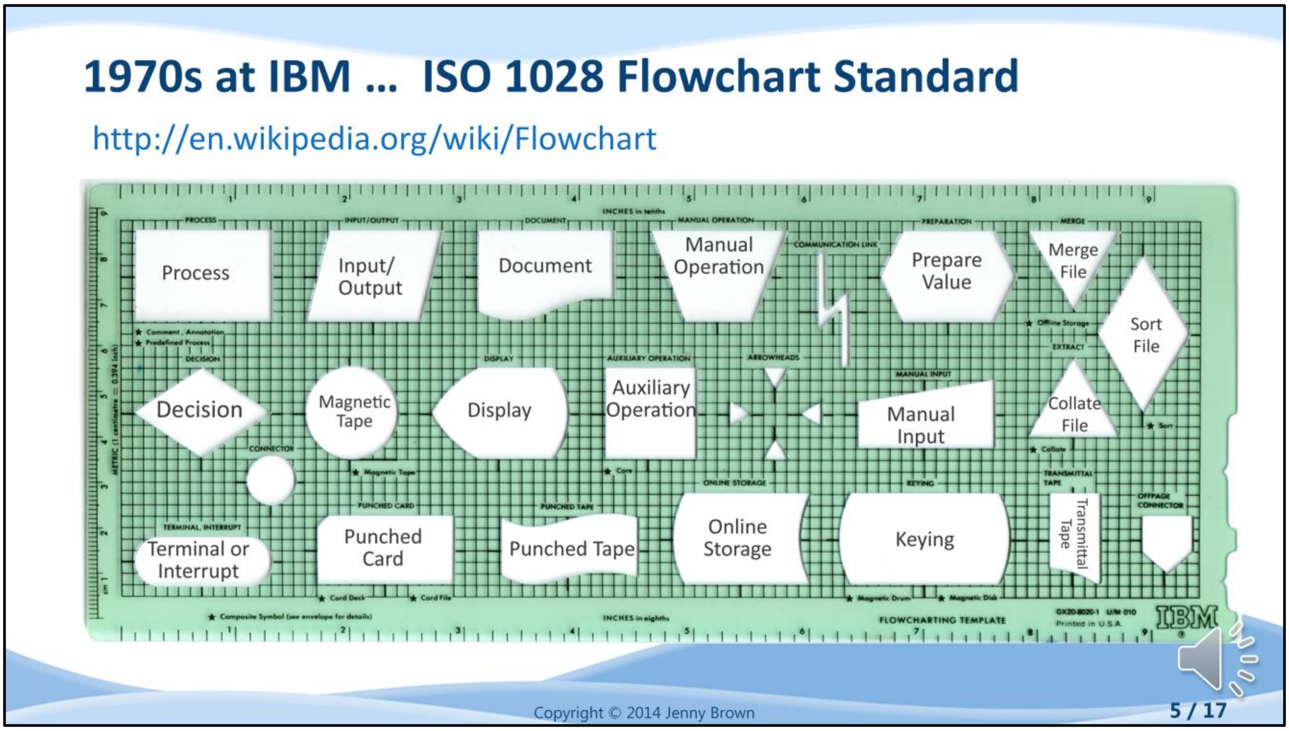
Process flow diagrams started in the 1920's, as factory engineers communicated about factory processes.



1970s at IBM … ISO 1028 Flowchart Standard

They gradually migrated into business use through the 40's and 50's, and then software planning in the 1970's.

Flowcharts were drawn by hand on paper, and plastic stencils made it easier to draw. They're small though, which meant any writing inside the box was pretty crowded. As a result, the text for each step was kept very short, and if more detail was needed, it was written to the side or in another flow diagram.

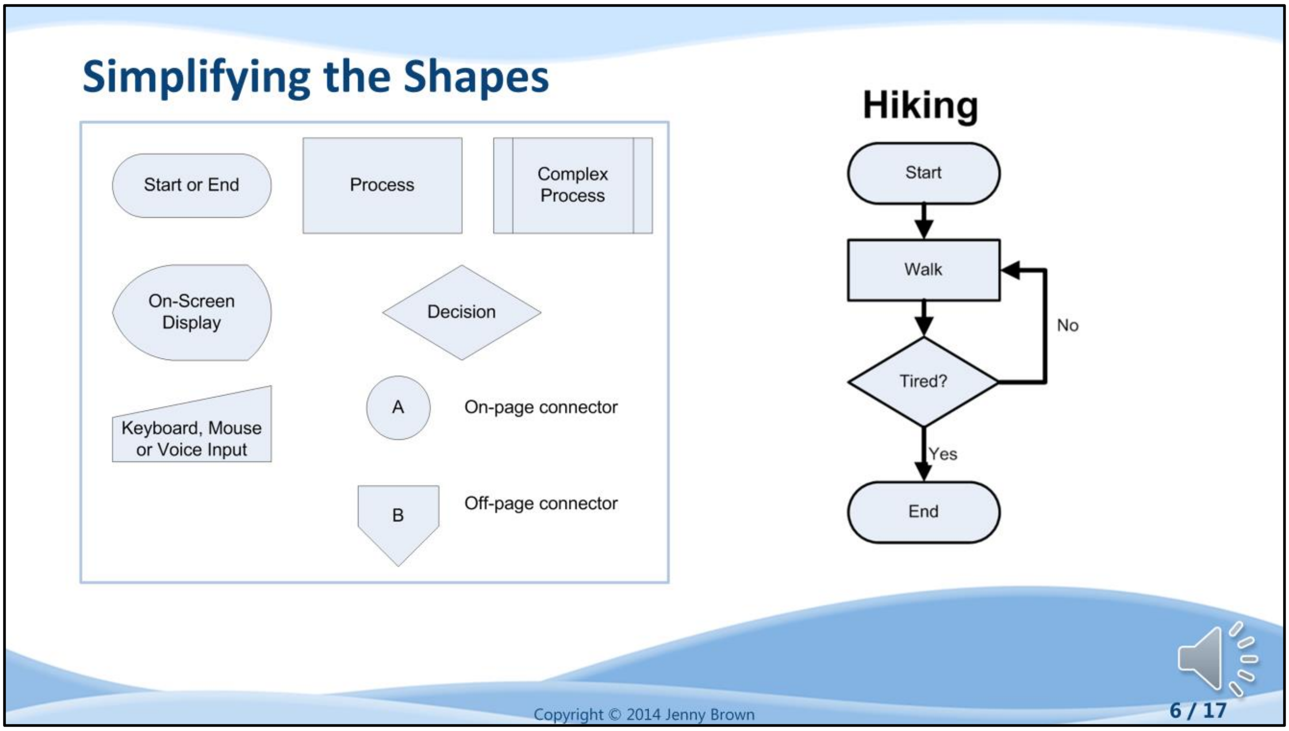


1970s at IBM … ISO 1028 Flowchart Standard

http://en.wikipedia.org/wiki/Flowchart

5 / 17

Here are the names of the shapes. You can see that some of them are no longer in use - for instance, we no longer store source code on punch cards or paper tape. Most developers won't recognize all of these symbols, so we're going to narrow down the set so it better fits modern programming. Check out the Wikipedia article for more details if you're curious about the rest.



Simplifying the Shapes

Start or End

Process

Complex Process

On-Screen Display

Decision

Keyboard, Mouse or Voice Input

A On-page connector

B Off-page connector

Hiking

Start

Walk

Tired?

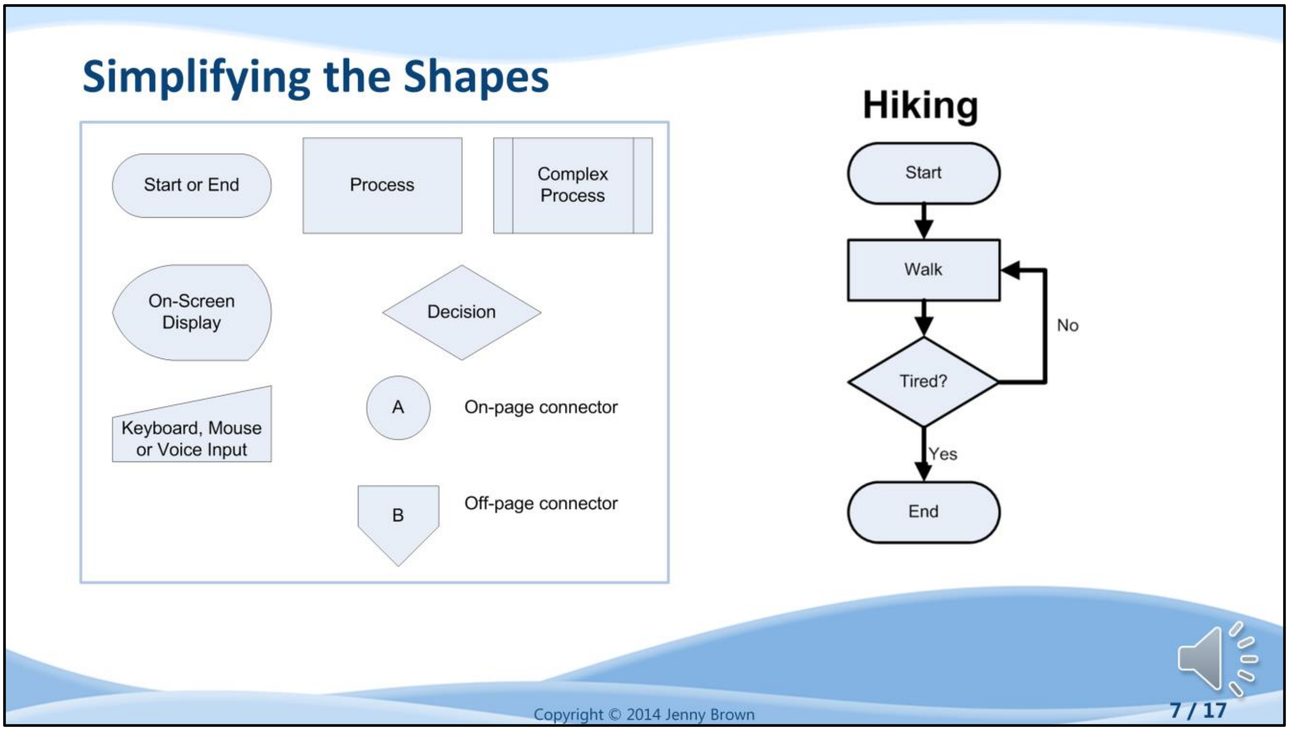
No

Yes

End

These shapes, plus hand-drawn arrows, will be plenty for the kinds of software we're working on in this course.

These are all fairly easy to sketch by hand. Don't worry if your drawings are a little bit sloppy. The concept is what matters, not the art.



Simplifying the Shapes

Start or End

Process

Complex Process

On-Screen Display

Decision

Keyboard, Mouse or Voice Input

A On-page connector

B Off-page connector

Hiking

Start

Walk

Tired?

No

Yes

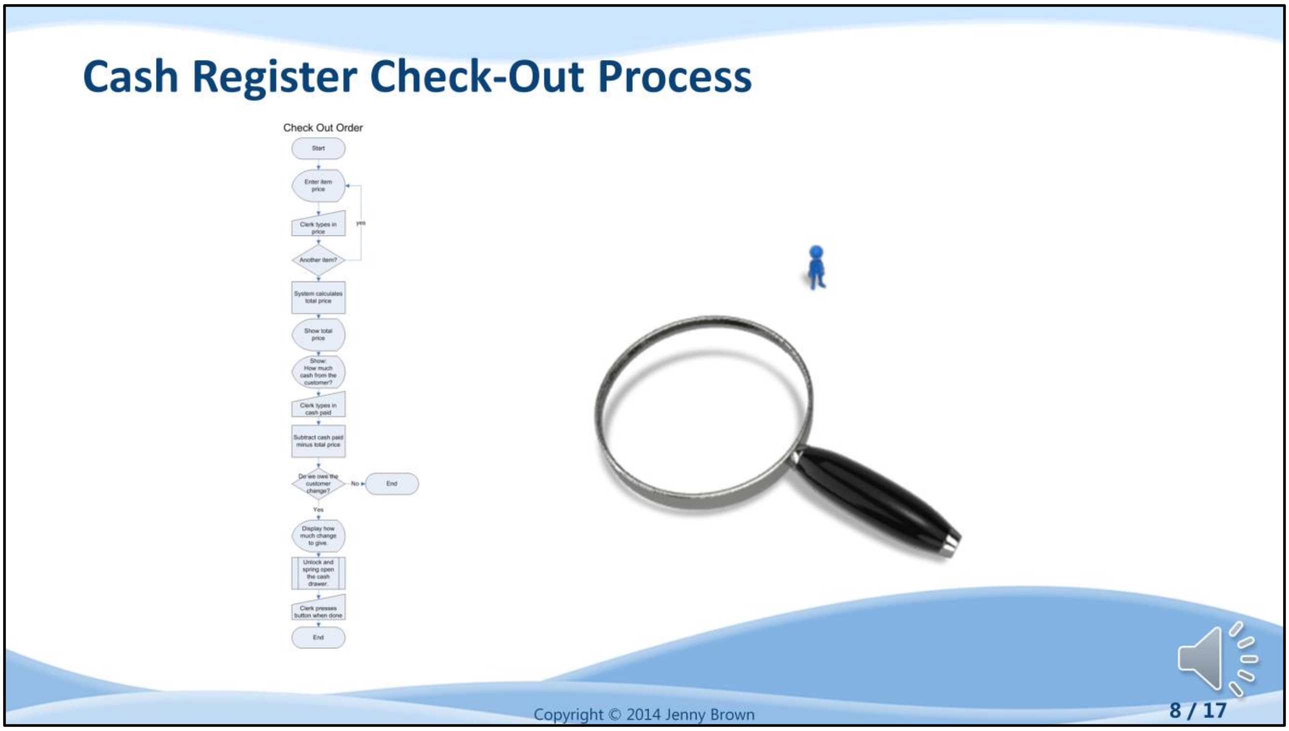
End

7 / 17

The "Start or End" rounded box is used to start each flow chart. Usually there is only one starting point, and this shows where that is. It's simply labeled "Start". There may be multiple ending points, and each end point should have the same rounded box simply marked "End".

The process box is the most commonly used shape. It describes a step in the process. If one step is actually many steps, it might use the Complex Process shape instead, which usually implies there's another flowchart somewhere that gives the details inside that. Decision boxes are like if statements - they have more than one branch, labeled with the choice that would send you down that branch.

Let's see a more complex flow chart, and we'll explore each of the shapes along the way.



Cash Register Check-Out Process

Check Out Order

Start

Enter item price

Clerk types in price

yes

Another item?

System calculates total price

Show total price

Show: How much cash from the customer?

Clerk types in cash paid

Subtract cash paid minus total price

Do we owe the customer change?

No

End

Yes

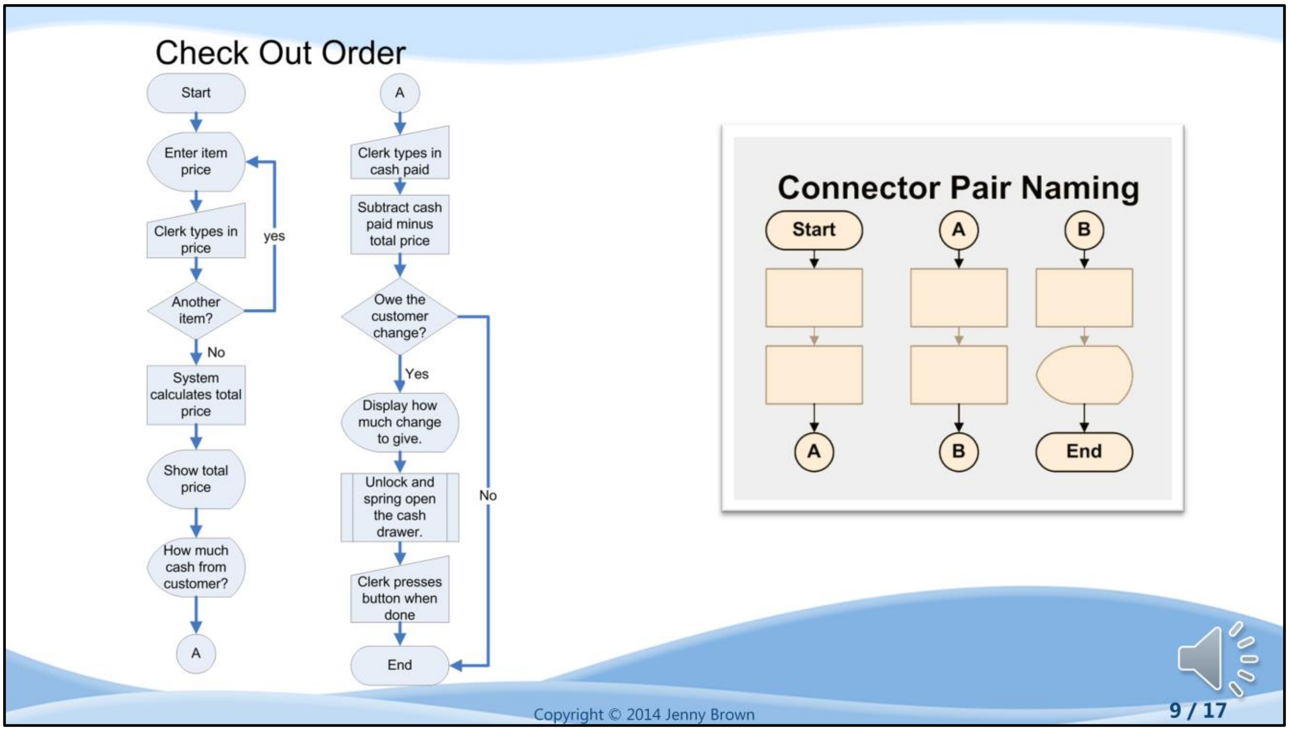
Display how much change to give.

Unlock and spring open the cash drawer.

Clerk presses button when done

End

Here's an example flow chart for a cash register check-out process. Hmm, that's kind of hard to read. We should break it into a couple of pieces that fit better, so let's see it in two parts with an on-page connector used instead.



Check Out Order

Start
Enter item price
Clerk types in price
Another item?
yes
No
System calculates total price
Show total price
How much cash from customer?
A

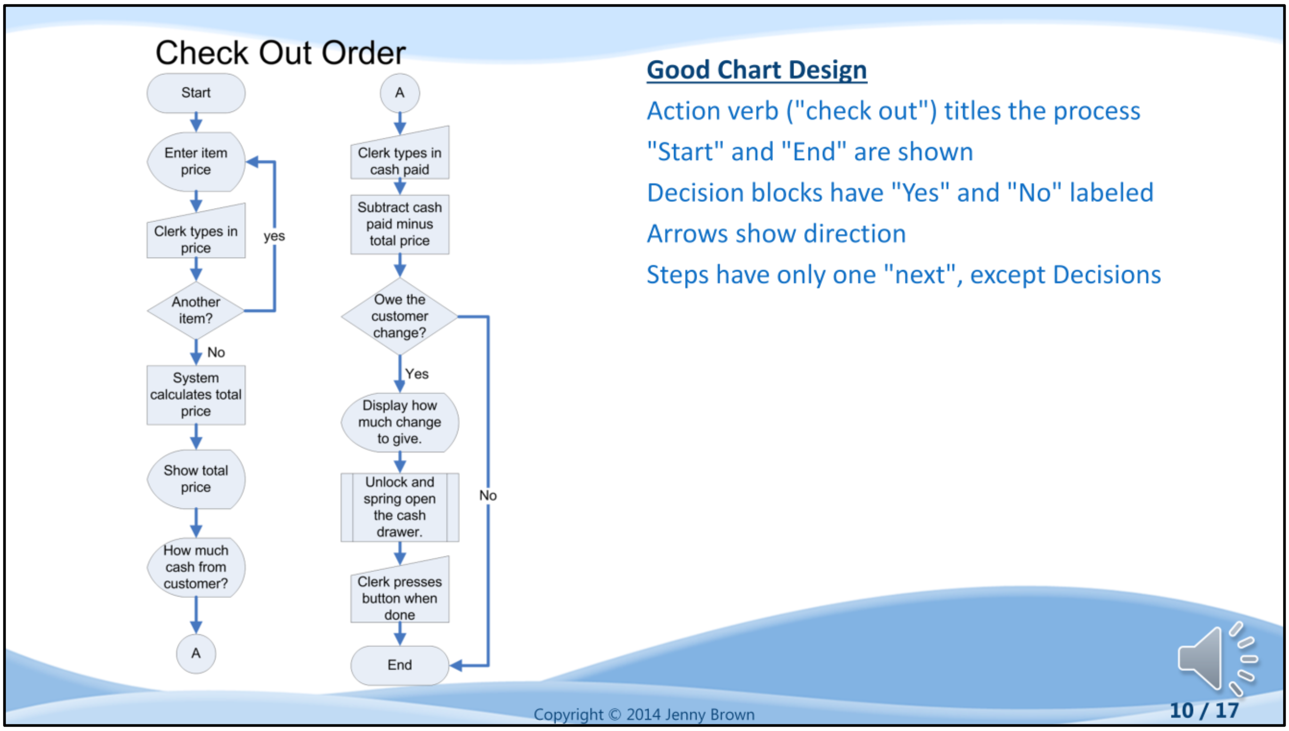
A
Clerk types in cash paid
Subtract cash paid minus total price
Owe the customer change?
Yes
Display how much change to give.
Unlock and spring open the cash drawer.
Clerk presses button when done
No
End

Connector Pair Naming

Start
A
B
A
B
End

9 / 17

Ahh, that's much more readable. Flowcharts start at the upper left, and move down and to the right. In this case, I used "A" to label my on-page connector pair, because it was the first one. If I had a second pair, I would label it "B", and a third pair would be labeled "C", and so on. You can use the same kind of naming for off-page connectors.


Check Out Order
Start
Enter item price
Clerk types in price
yes
Another item?
No
System calculates total price
Show total price
How much cash from customer?
A
A
Clerk types in cash paid
Subtract cash paid minus total price
Owe the customer change?
Yes
Display how much change to give.
Unlock and spring open the cash drawer.
Clerk presses button when done
No
End

Good Chart Design

Action verb ("check out") titles the process

"Start" and "End" are shown

Decision blocks have "Yes" and "No" labeled

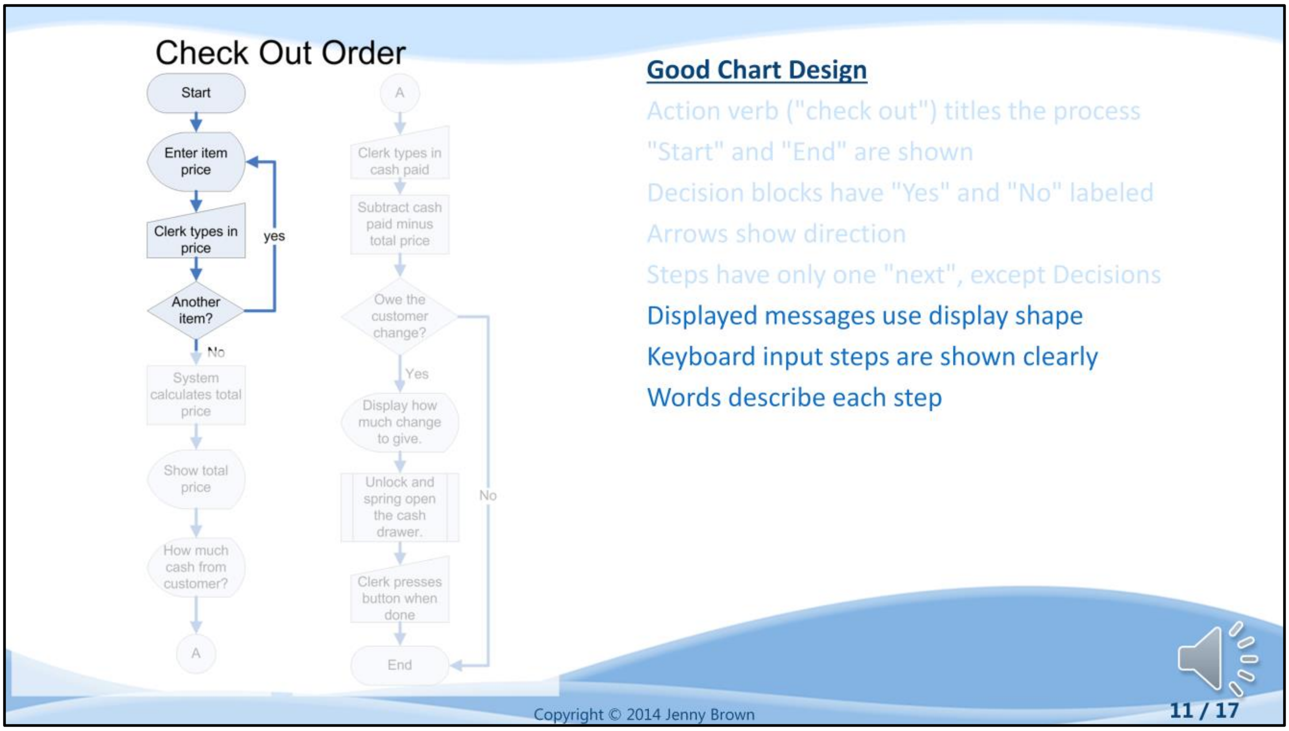
Arrows show direction

Steps have only one "next", except Decisions

10 / 17

Let's walk through this flow together. First we see the title above the chart, "Check Out Order". This is an action verb phrase, that tells us what process the flow chart is mapping out. When you have several flow charts for a single program, it's very helpful to name each of them. We also see the "Start" block right there at the top left.

Notice that most blocks have only one arrow coming out and going to the next block. The only exception is the decision blocks, which can have multiple choices. So your flow charts should have only one arrow coming out of each box unless it's a decision block with a question in it.


Check Out Order
Start
Enter item price
Clerk types in price
yes
Another item?
No
System calculates total price
Show total price
How much cash from customer?
A
A
Clerk types in cash paid
Subtract cash paid minus total price
Owe the customer change?
Yes
Display how much change to give.
Unlock and spring open the cash drawer.
No
Clerk presses button when done
End

Good Chart Design

Action verb ("check out") titles the process

"Start" and "End" are shown

Decision blocks have "Yes" and "No" labeled

Arrows show direction

Steps have only one "next", except Decisions

Displayed messages use display shape

Keyboard input steps are shown clearly

Words describe each step

11 / 17

Now let's focus in on the first few blocks. First we're using the display symbol, to show the clerk a message "Enter item price". This is basically a print statement; it displays some text on the clerk's screen. Since we're just making the flowchart right now, we don't have to worry about exactly how it shows up, or the wording of the message, or what fonts or colors it's in. All we care about is that the computer tells the sales clerk that they should enter the item price.

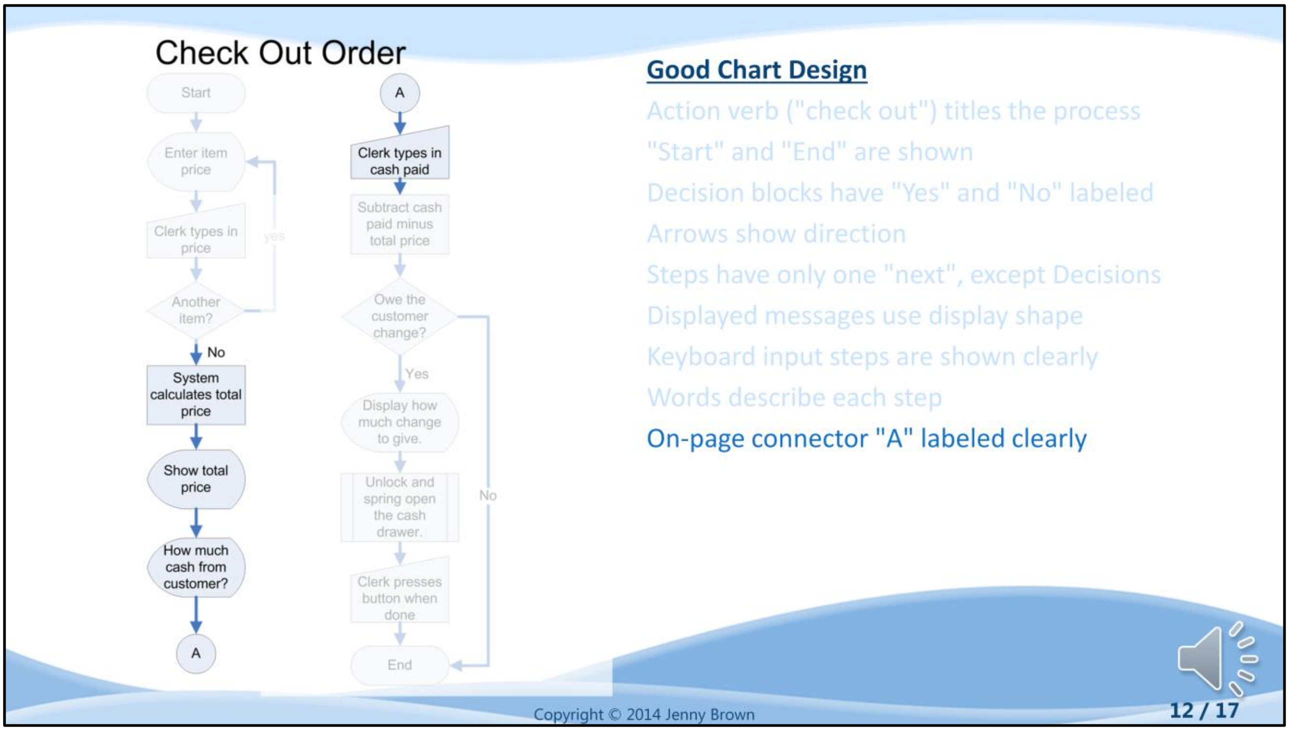
The next block is the manual input block. We've labelled it, "Clerk types in price". Our program will have to sit and wait for them to type it in.

Then the program makes a decision; it asks the question, will the clerk be entering another item? Now, again we don't really care how we know this. Maybe the clerk pushes a different button when they enter the last item. Or maybe we have to display a prompt and have them answer yes or no. We're not really worried about that detail right now. All we care about is somehow knowing whether there's another item coming or not.

If there is another item, then we need to ask for its price and get input again. So we loop back to the earlier step to repeat some steps. I haven't shown you how to write that in source code yet, but we'll get there.

So if there was another item, we keep asking for prices until we've got them all.

Then, "No", there's not another item, so then we continue on with the flow chart.


Check Out Order
Start
Enter item price
Clerk types in price
yes
Another item?
No
System calculates total price
Show total price
How much cash from customer?
A
A
Clerk types in cash paid
Subtract cash paid minus total price
Owe the customer change?
Yes
Display how much change to give.
Unlock and spring open the cash drawer.
No
Clerk presses button when done
End

Good Chart Design

Action verb ("check out") titles the process

"Start" and "End" are shown

Decision blocks have "Yes" and "No" labeled

Arrows show direction

Steps have only one "next", except Decisions

Displayed messages use display shape

Keyboard input steps are shown clearly

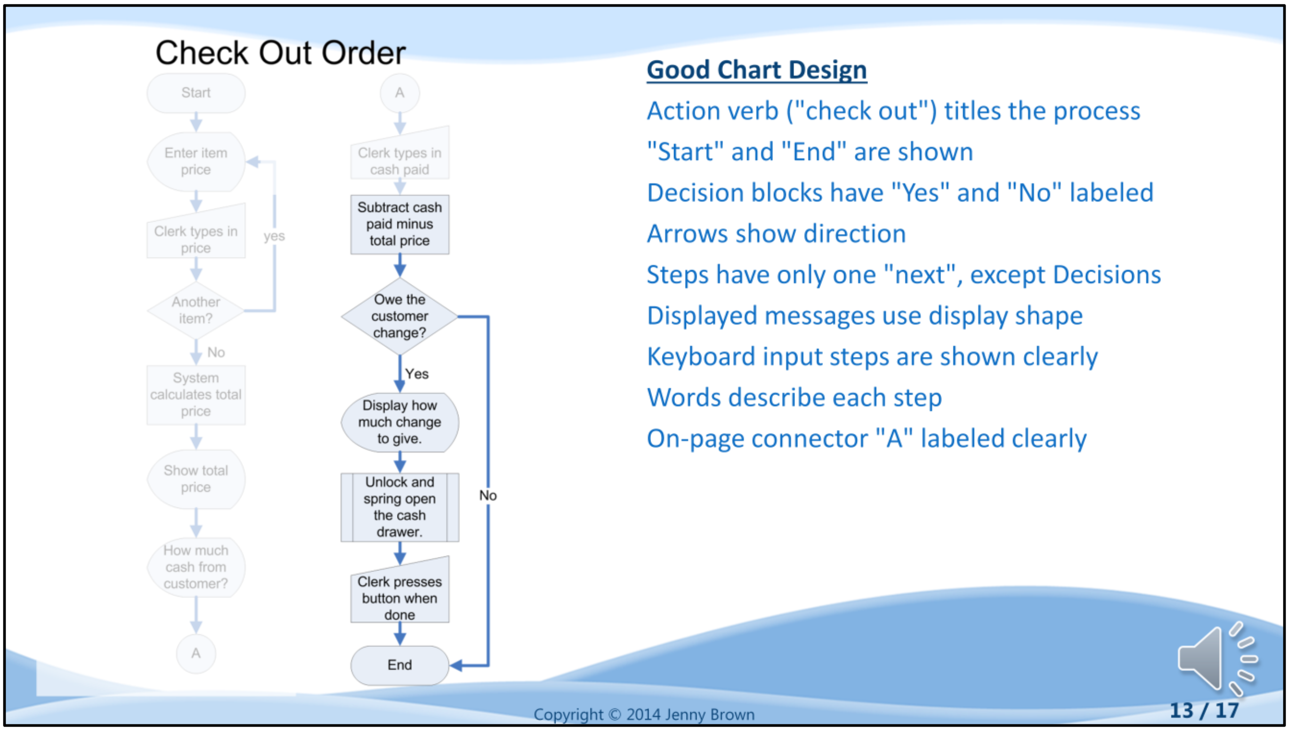
Words describe each step

On-page connector "A" labeled clearly

12 / 17

Now the clerk has entered all the prices. So the computer system needs to calculate the total price for the whole order, by adding them up. Then it shows the total price on screen, so the clerk can tell the customer what the total is.

Next, the system asks the clerk to please enter how much cash the customer paid with. First it displays the question, and then it waits for the clerk to type in the amount of cash paid. Remember here to follow the on-page connector "A" to the next part of the flow chart.



Check Out Order

Start
Enter item price
Clerk types in price
Another item?
yes
No
System calculates total price
Show total price
How much cash from customer?
A

A
Clerk types in cash paid
Subtract cash paid minus total price
Owe the customer change?
Yes
Display how much change to give.
Unlock and spring open the cash drawer.
Clerk presses button when done
No
End

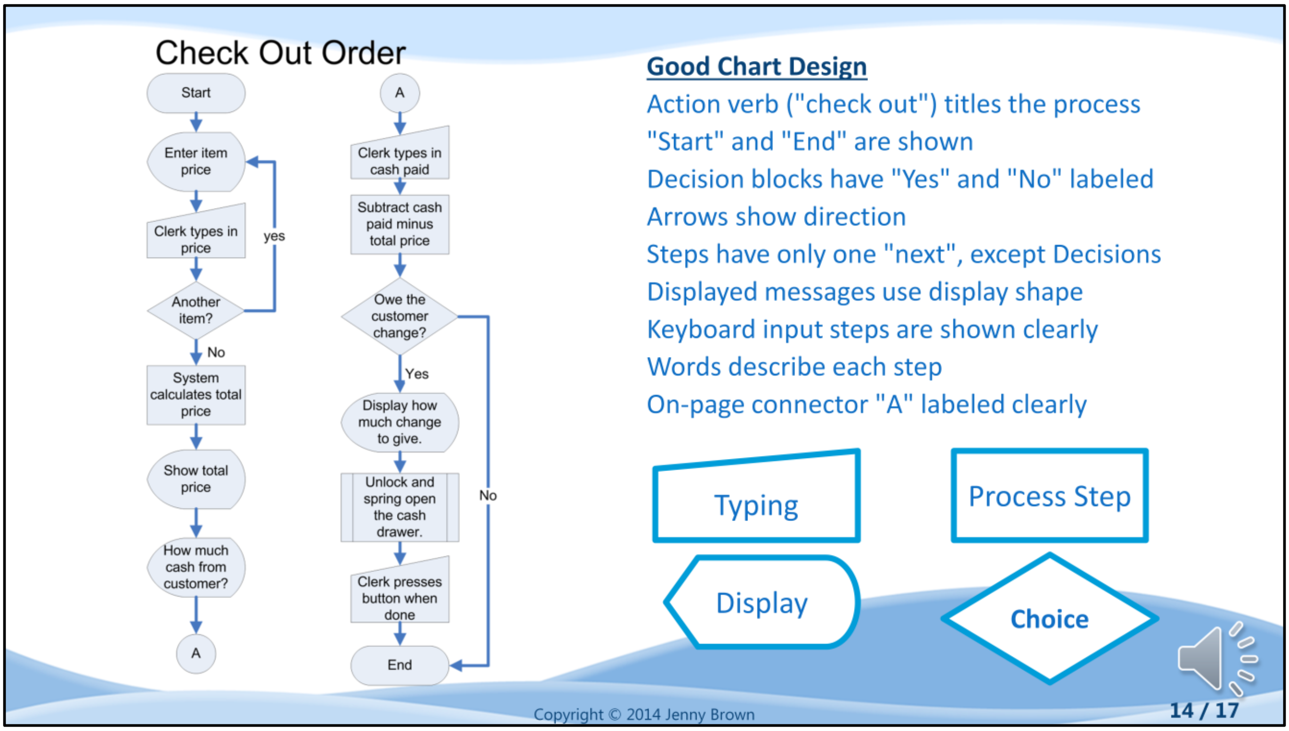
Good Chart Design

Action verb ("check out") titles the process
"Start" and "End" are shown
Decision blocks have "Yes" and "No" labeled
Arrows show direction
Steps have only one "next", except Decisions
Displayed messages use display shape
Keyboard input steps are shown clearly
Words describe each step
On-page connector "A" labeled clearly

13 / 17

Next there's a process block. The system calculates cash paid minus order price. Then it goes to a decision block. If the order price and cash paid were exactly matched, then we don't owe the customer any change, so we're done - End. If we do owe change, display on the screen how much change we should give.

Then the system automatically unlocks and opens the cash drawer so the clerk can make change. This one is using a Complex Process block, implying that there are probably several steps involved in unlocking and opening the cash drawer. In any case, it's open now, so we move on to the next block. The clerk presses a button on the system to indicate that they're done making change (or maybe they just close the cash drawer) and that finishes the process. We finish off with an "End" block.


Check Out Order
Start
Enter item price
Clerk types in price
yes
Another item?
No
System calculates total price
Show total price
How much cash from customer?
A
A
Clerk types in cash paid
Subtract cash paid minus total price
Owe the customer change?
Yes
Display how much change to give.
Unlock and spring open the cash drawer.
No
Clerk presses button when done
End
Typing
Process Step
Display
Choice

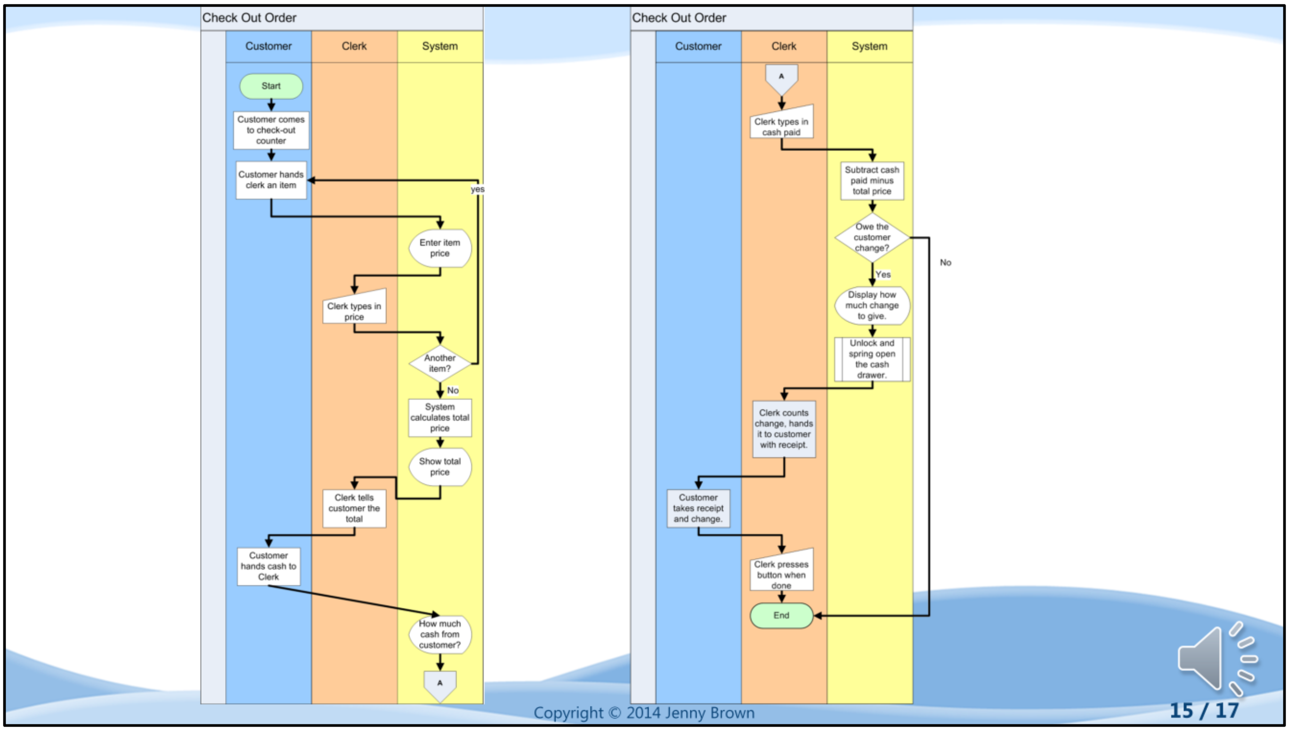
Good Chart Design

Action verb ("check out") titles the process
"Start" and "End" are shown
Decision blocks have "Yes" and "No" labeled
Arrows show direction
Steps have only one "next", except Decisions
Displayed messages use display shape
Keyboard input steps are shown clearly
Words describe each step
On-page connector "A" labeled clearly

14 / 17

Notice how every time the clerk types something in, we're using the manual input shape. <click>
And every time the system shows something on screen, we're using the display shape. <click>
When the system is doing a calculation, that doesn't need input or output, it's just a process step in a process block. <click>
When we need to make a decision, we use the diamond, and label the arrows with the choices. <click>

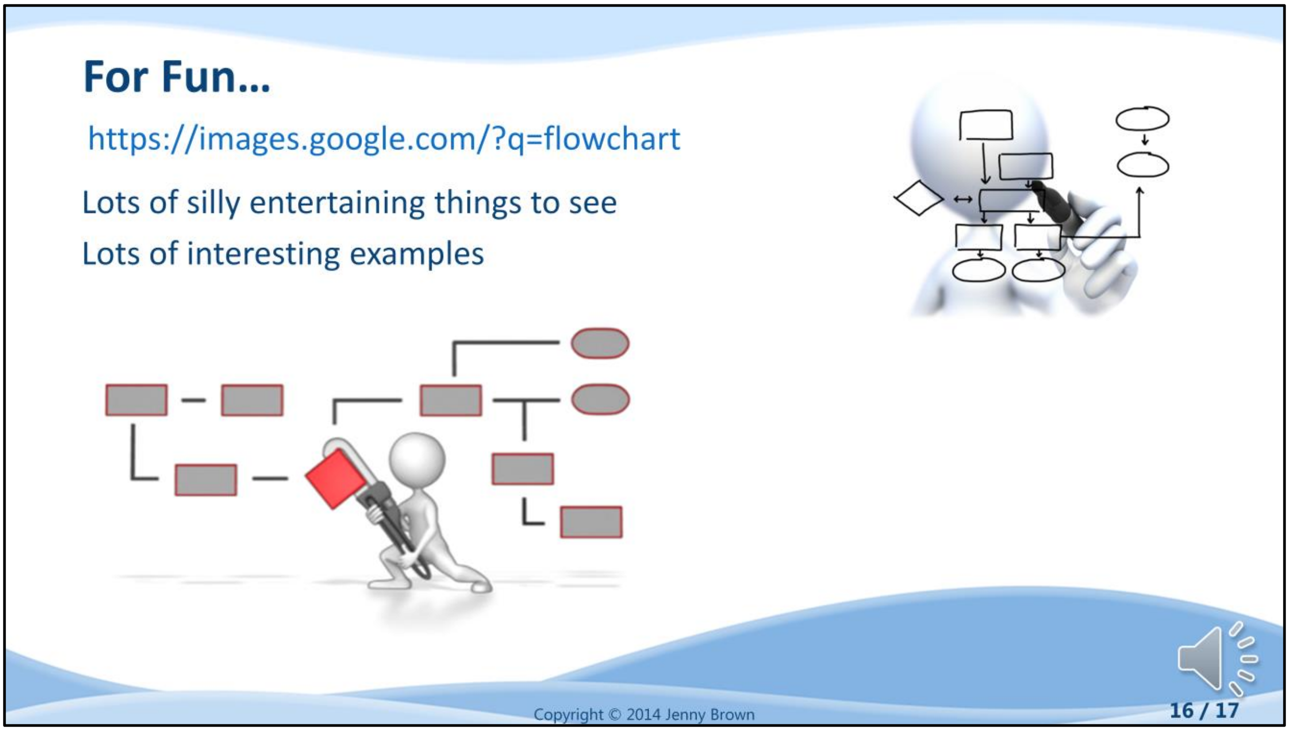
When you use a flow chart to plan your program, it makes you think step-by-step, carefully. Deciding what shapes to use for each block forces you to use small steps which better match how software works. You don't have to worry about coding syntax at this stage. It's just a planning step to make sure you understand, in human language, what the program needs to do.


Check Out Order
Customer
Clerk
System
Start
Customer comes to check-out counter
Customer hands clerk an item
yes
Enter item price
Clerk types in price
Another item?
No
System calculates total price
Show total price
Clerk tells customer the total
Customer hands cash to Clerk
How much cash from customer?
A
Check Out Order
Customer
Clerk
System
A
Clerk types in cash paid
Subtract cash paid minus total price
Owe the customer change?
No
Yes
Display how much change to give.
Unlock and spring open the cash drawer.
Clerk counts change, hands it to customer with receipt.
Customer takes receipt and change.
Clerk presses button when done
End
Copyright © 2014 Jenny Brown
15 / 17

There's another variation on the flow chart, called a swim-lane flow chart. This uses the same shapes, but it clarifies who performs each step. The people or teams involved in the process each have their own color stripe, and the flow chart moves back and forth between them depending on who is performing that step.

Swim lane charts are especially useful for describing business processes that happen across several departments in a company. They can be used to figure out if there's a step in the process where a particular team isn't involved, or a step where a particular team is overloaded, to find ways to improve how the whole team works. They also clarify where automated systems are stuck waiting for a person to press the "Approve" button, and maybe that step could be eliminated to make a process faster.

Diagrams like this are used in many different ways. We've discussed using flow charts to help you plan software processes, but they really can be used for any process at all.



For Fun…

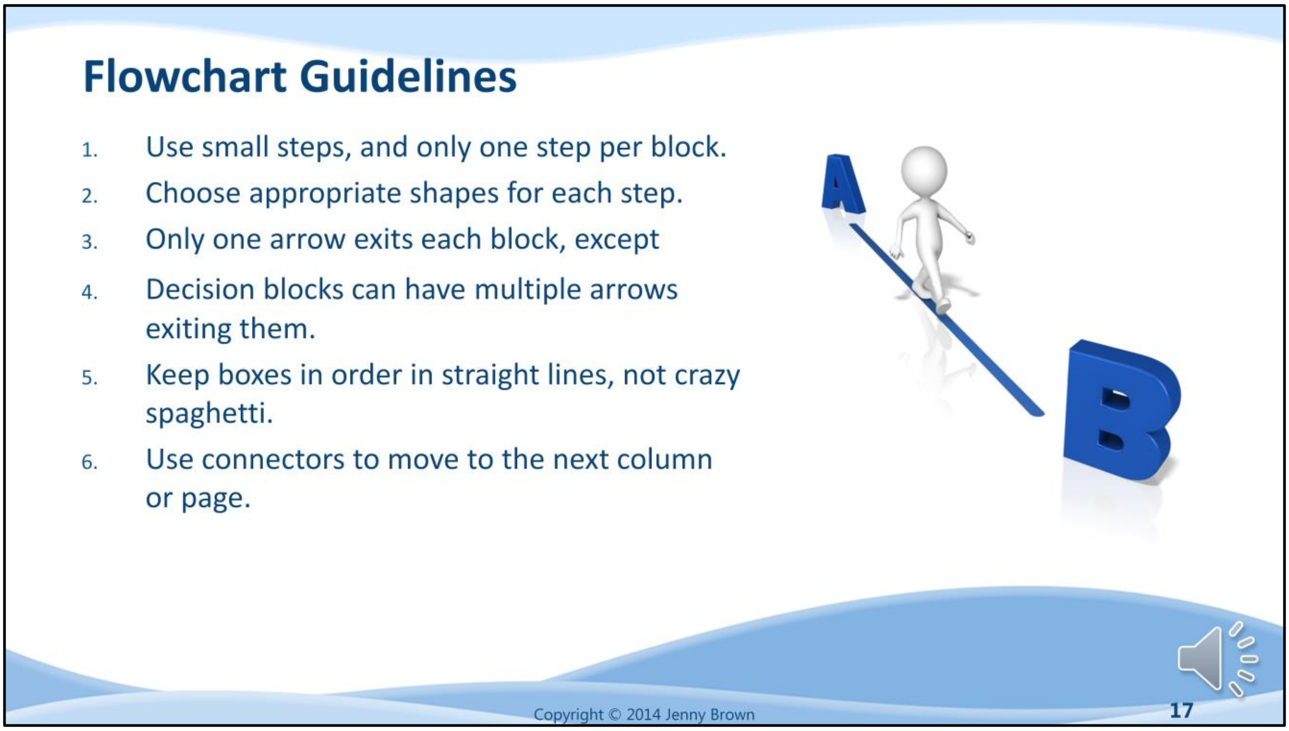
https://images.google.com/?q=flowchart

Lots of silly entertaining things to see

Lots of interesting examples

16 / 17

Flow charts are used for all kinds of things, including jokes. If you want a fun distraction, do a google image search for the word flowchart. There are all kinds of crazy and informative charts out there. You'll get a chance to see the diversity of styles and approaches, and also the similarities.

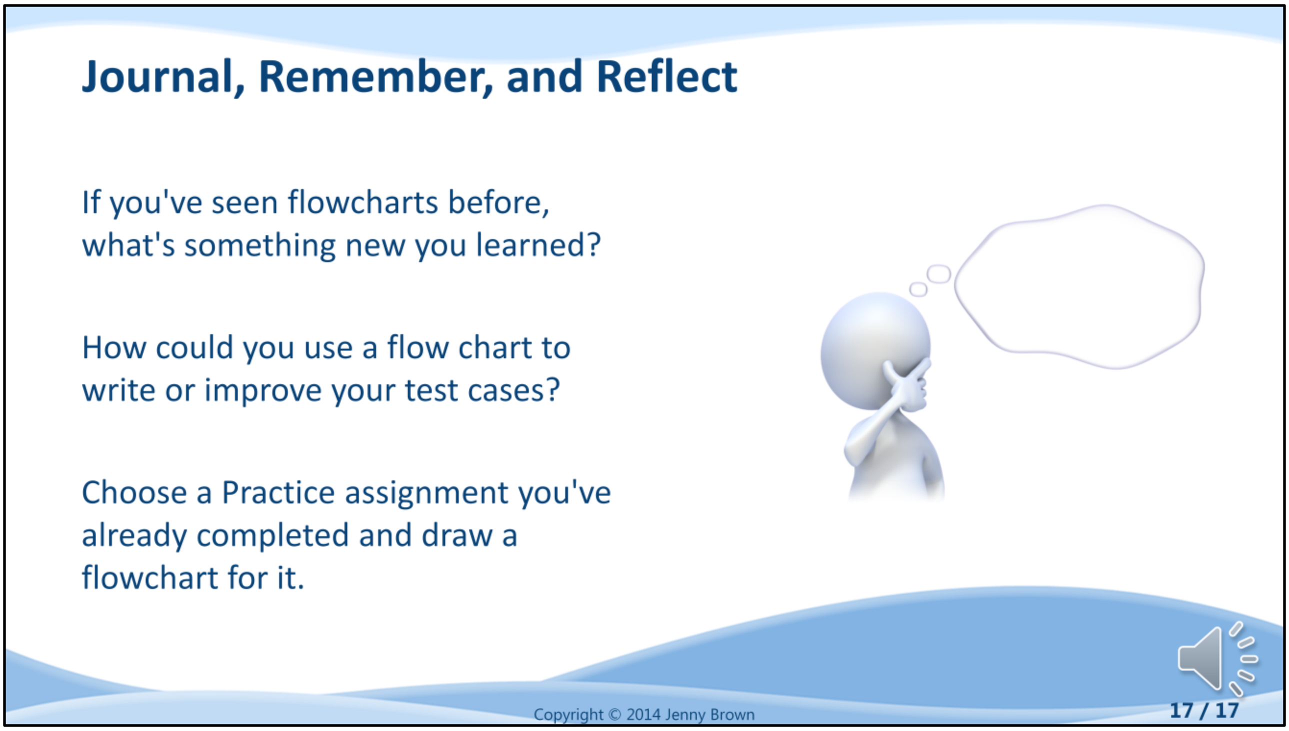


Flowchart Guidelines

1. Use small steps, and only one step per block.
2. Choose appropriate shapes for each step.
3. Only one arrow exits each block, except
4. Decision blocks can have multiple arrows exiting them.
5. Keep boxes in order in straight lines, not crazy spaghetti.
6. Use connectors to move to the next column or page.

Here are the "rules" or guidelines for drawing good flowcharts. Following these guidelines will make it easier to write source code that matches your flow chart, and it will make it easier for other people to understand your diagrams.

Use small steps, and only one step per block.
Choose appropriate shapes for each step.
Only one arrow exits each block, except
Decision blocks can have multiple arrows exiting them.
Keep boxes in order in straight lines, not crazy spaghetti, and
Use connectors to move to the next column or page.



Journal, Remember, and Reflect

If you've seen flowcharts before, what's something new you learned?

How could you use a flow chart to write or improve your test cases?

Choose a Practice assignment you've already completed and draw a flowchart for it.

Journal time. If you've seen flowcharts before, what's something new you learned from this video?

How could you use a flow chart to write or improve your test cases?

Choose a Practice assignment you've already completed and draw a flowchart for it.