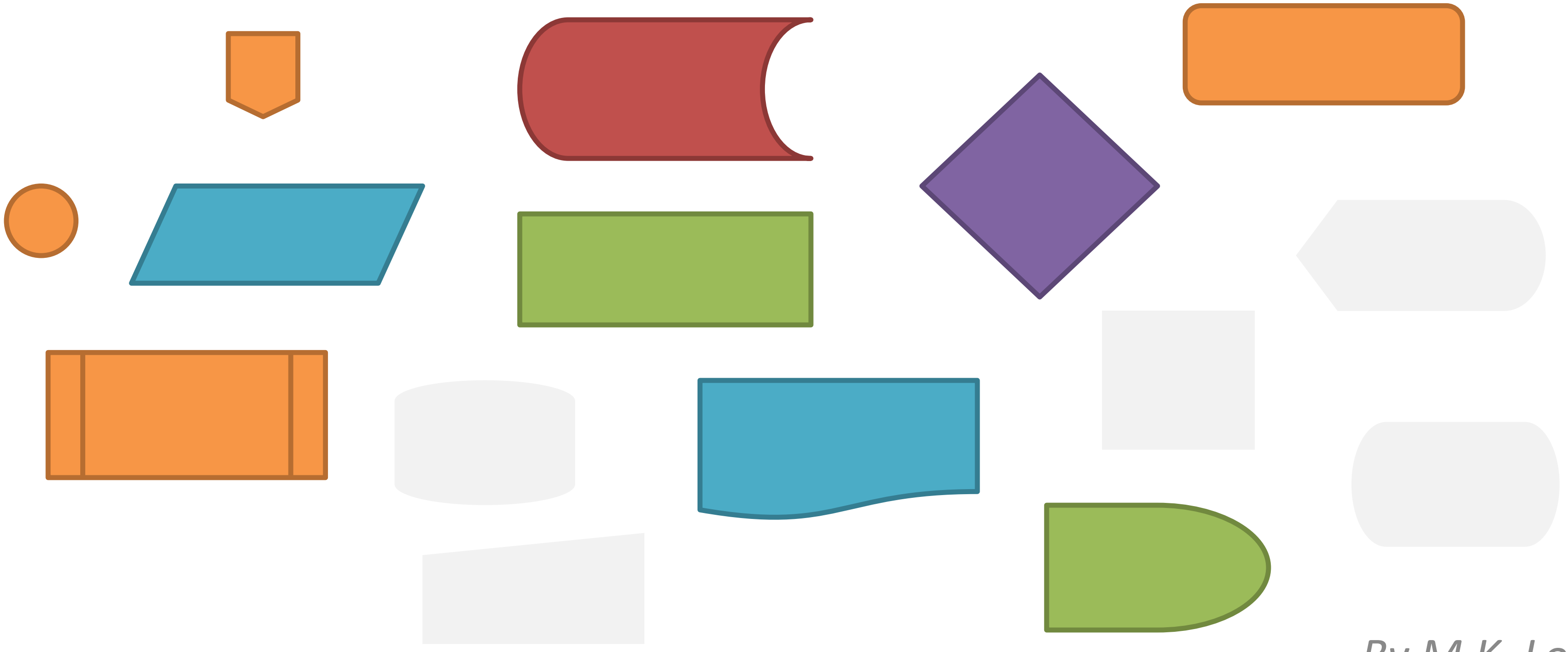
FLOWCHARTS

Part 2

IMPROVED VERSION!



By M.K. Lowe

Preface

Before using this guide, you must be completely familiar with (and understand) all of the content from guide 1. This guide builds upon much of the first guide, and assumes you understand all of its contents.

Please note where you see this in any examples:

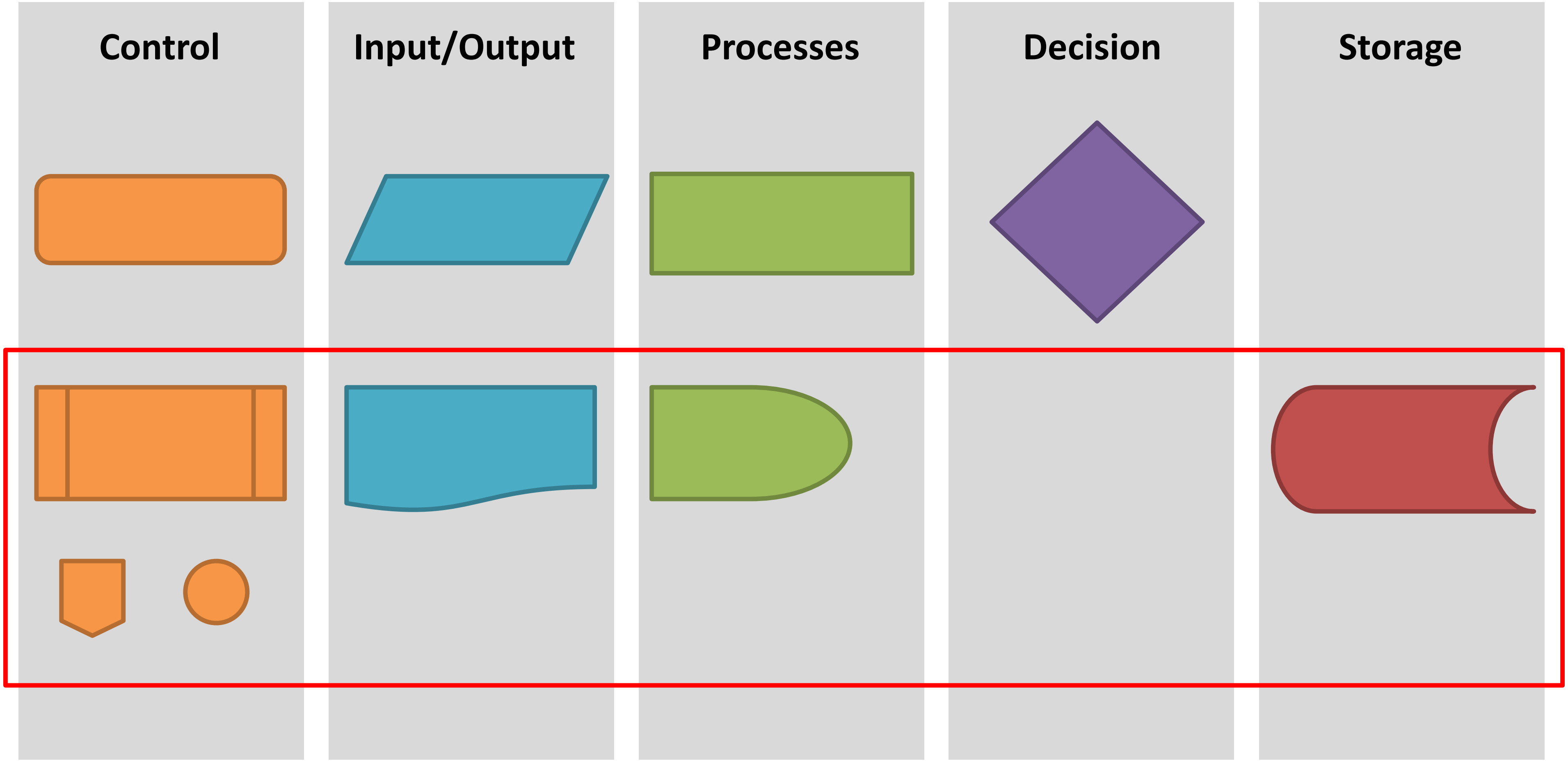
↓

....

It denotes that unimportant additional steps have been omitted in order to conserve space.

New Shapes

It is now time to look at some new shapes. The shapes are technically related, each with a more specific function:



Control

Input/Output

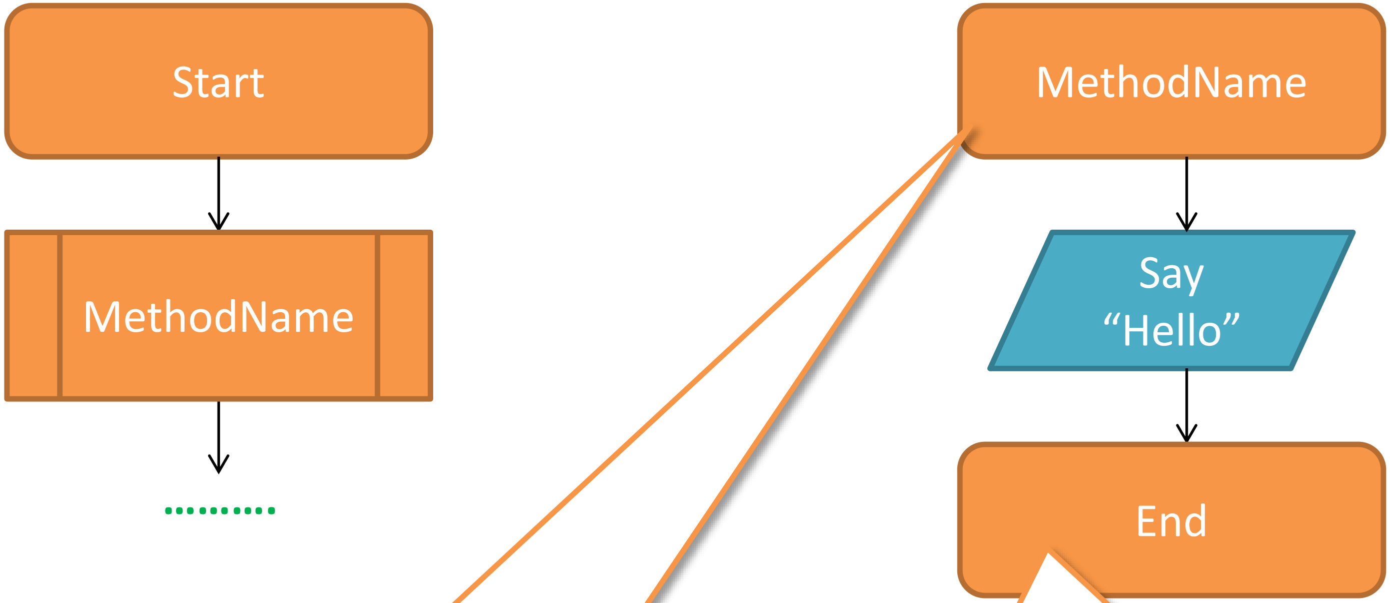
Processes

Decision

Storage

Sub-routine

Sub-routine allows us to call another method or function inside of our current algorithm. This is effectively used to reduce duplication of processes that frequently appear.

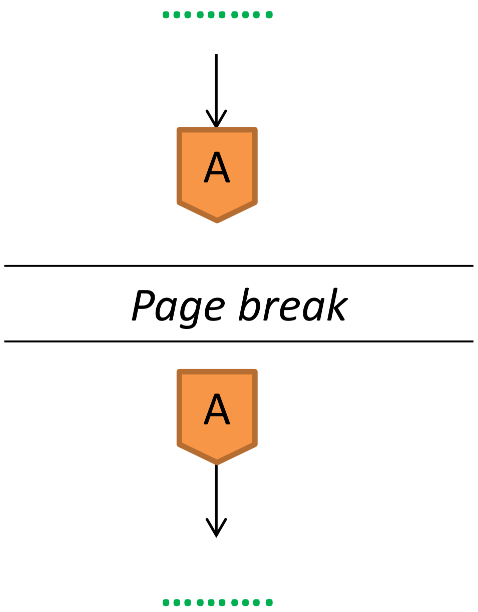


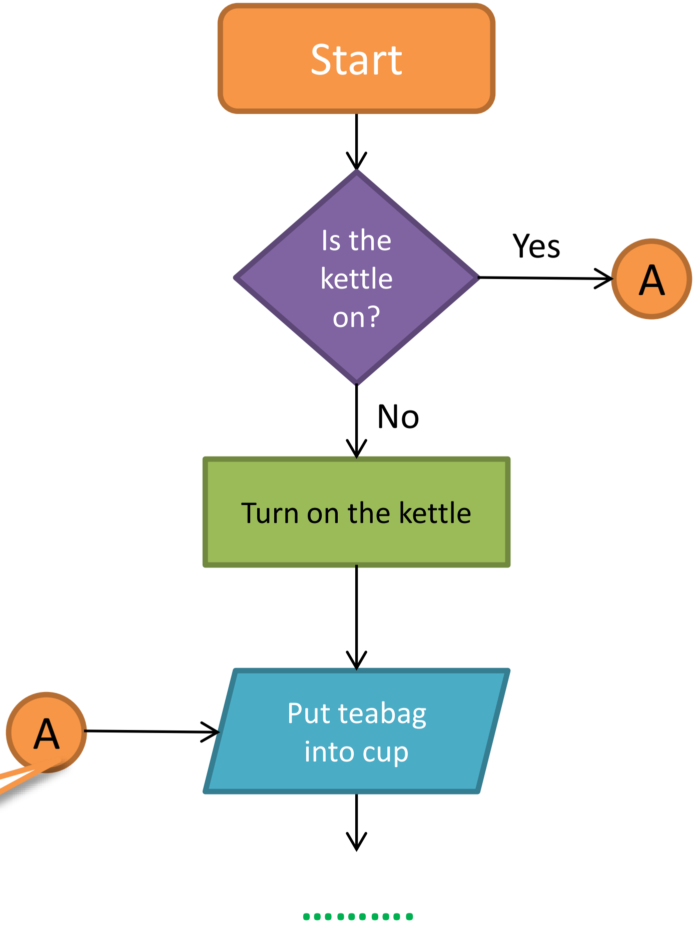
The subroutine is indicated by replacing START with the exact name of the routine being called.

When End is reached, we will return to the original flowchart and continue past the subroutine call.

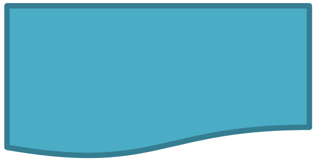
Off-page and On-page

These two icons helpfully indicate continuation of flowcharts. Use the **Off-page indicator** (pictured left) to show where a flowchart ends/starts on a new page. Use the **On-page indicator** (pictured right) to connect logic on the same page.

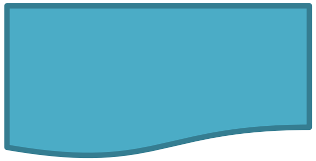




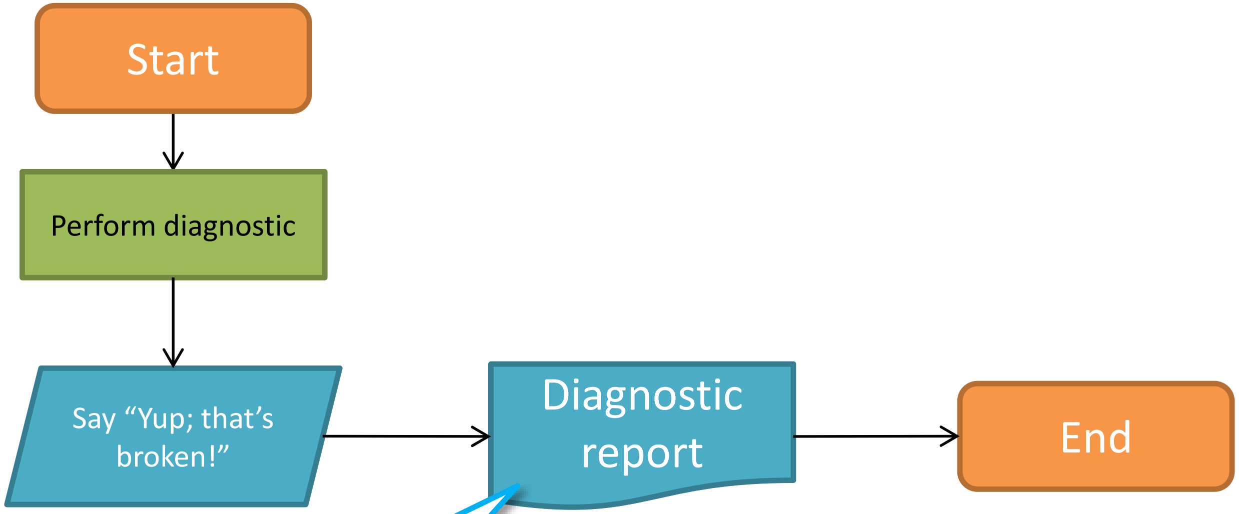
Letters or numbers are used to show which two are connected. Here, shapes A are connected.



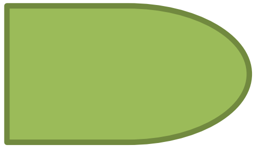
Document



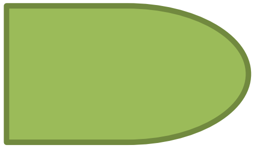
This denotes the production of a **document** at this point of the flowchart.



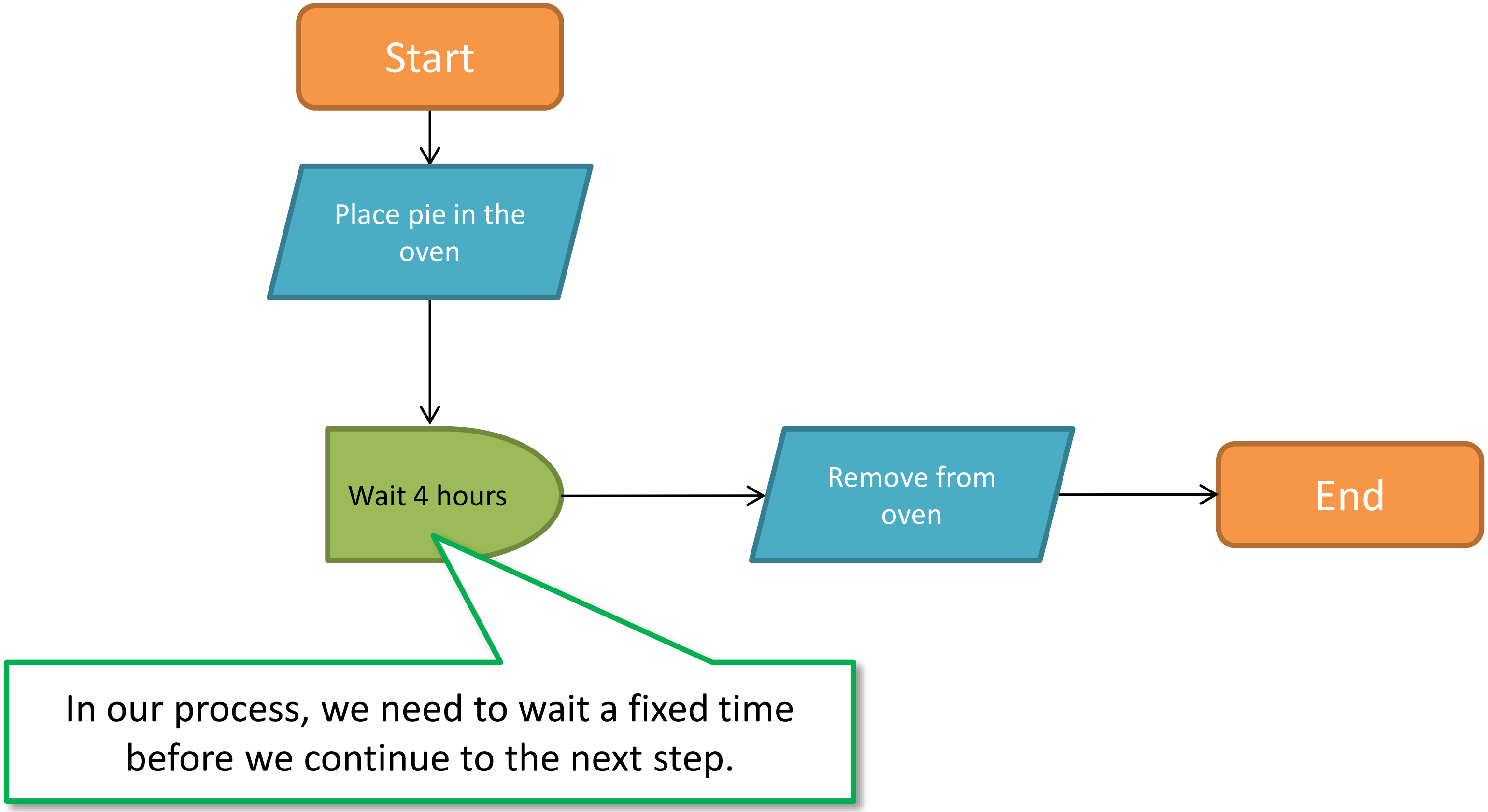
The name of the document that we are producing goes inside the shape. Here it is called “Diagnostic report”.



Delay



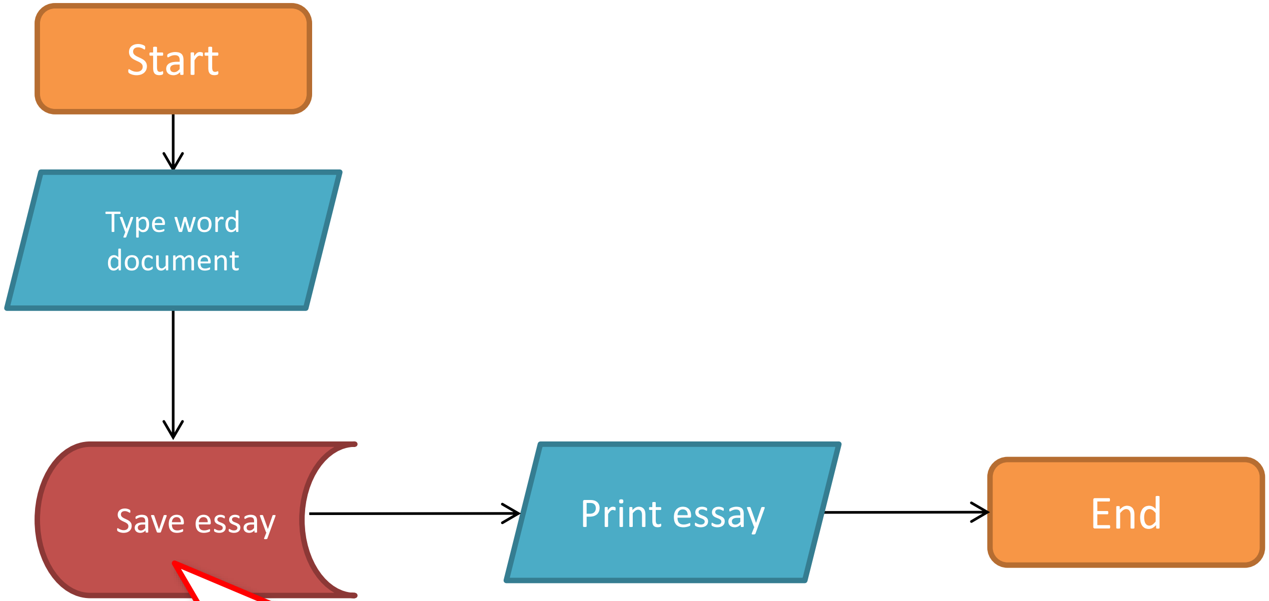
The **delay** symbol denotes we must wait for something to happen. It should be a fixed time period. Be careful not to set conditions and confuse it with a decision!



In our process, we need to wait a fixed time before we continue to the next step.

Data Storage

This is a generic **data storage** function. Here we indicate something is being saved somewhere. In later guides, we will look at how to make this more specific.



Start

Type word document

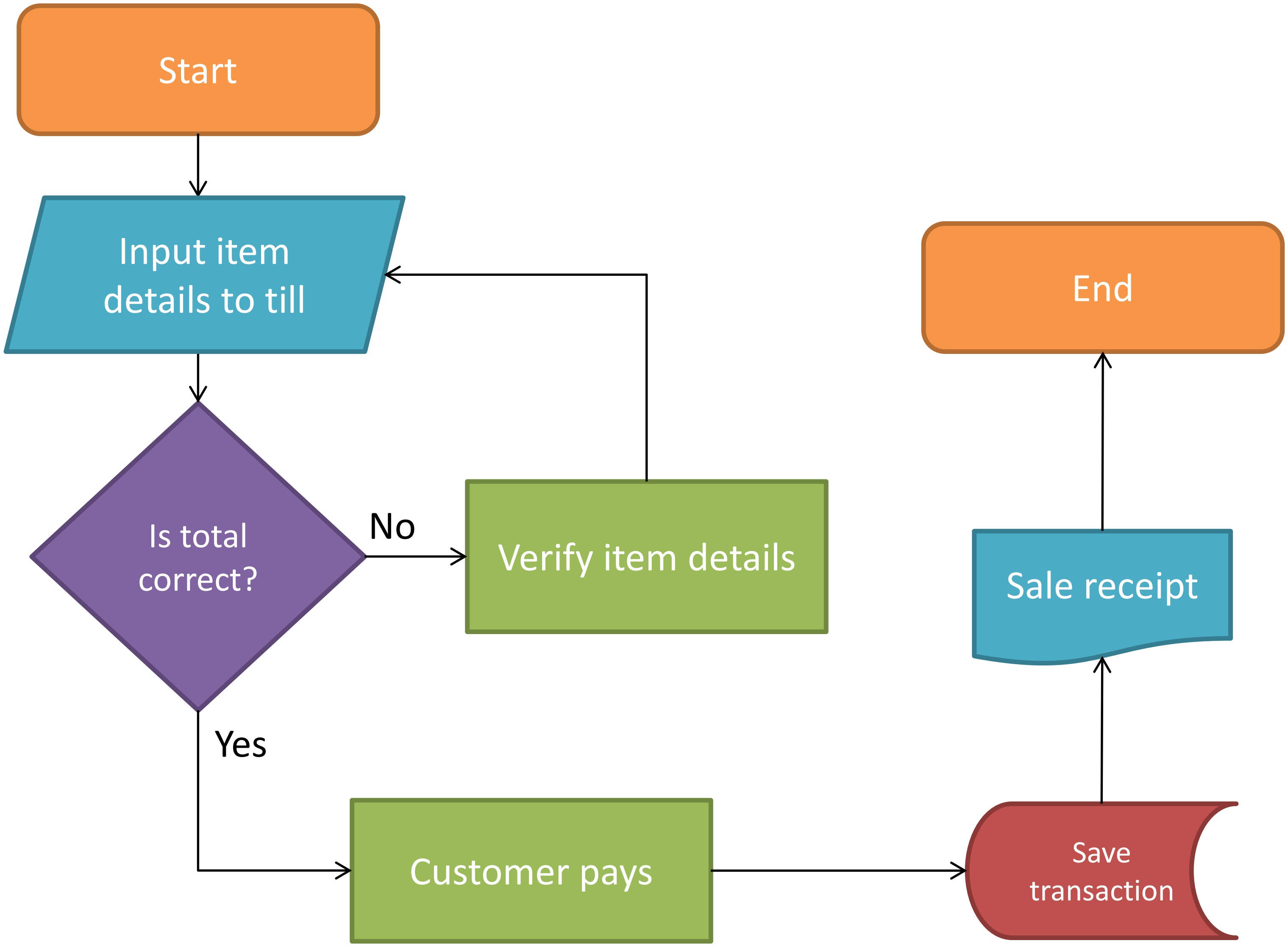
Save essay

Print essay

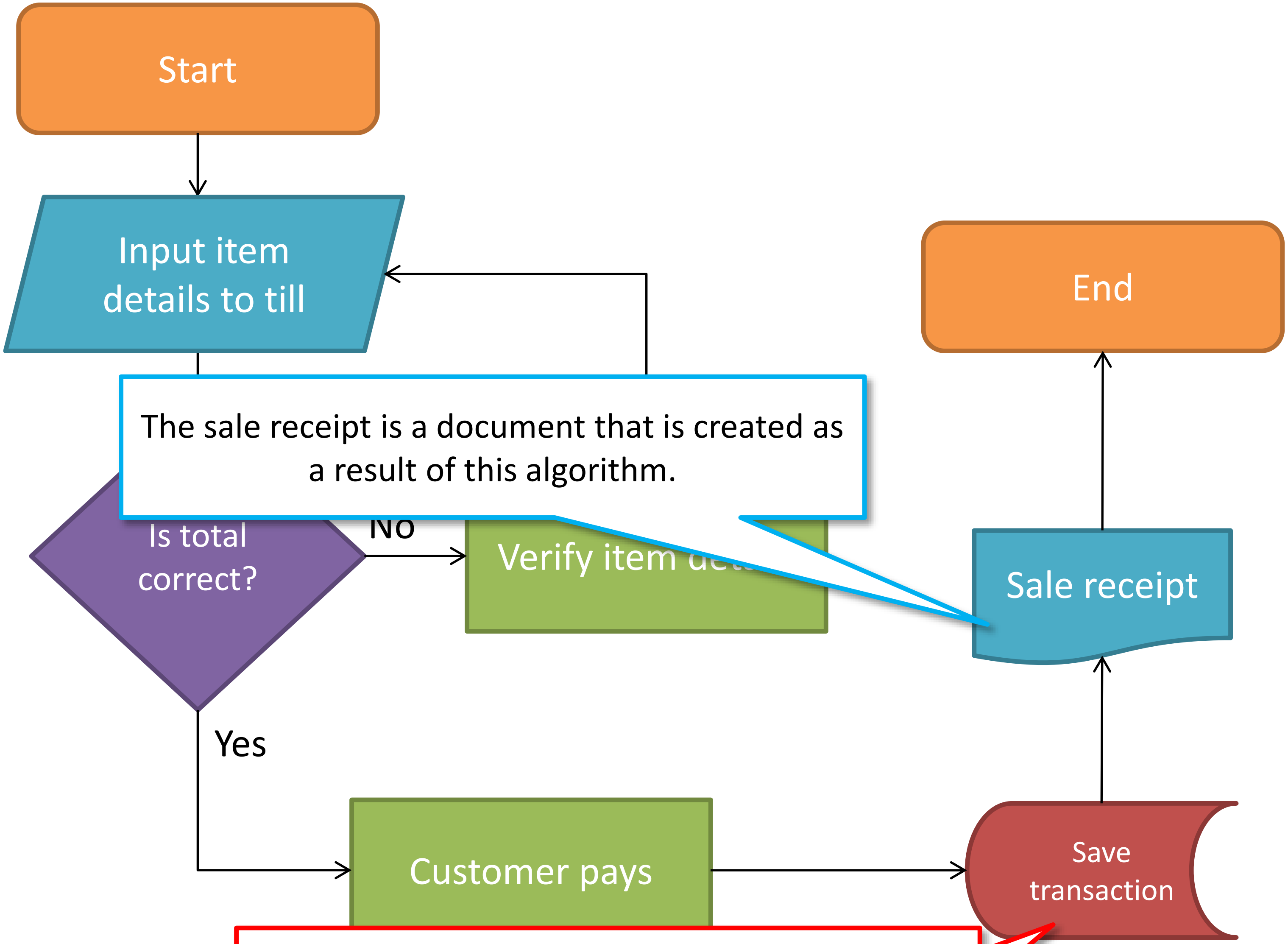
End

We simply indicate that we plan to save something. At this stage, we do not need to indicate how or where.

Example 1 – *Complete a sale in a shop.*



Example 1 – *Complete a sale in a shop.* Explained

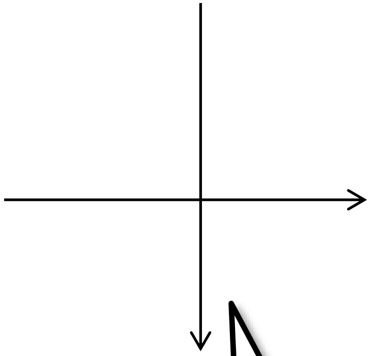


The sale receipt is a document that is created as a result of this algorithm.

The entire transaction obviously needs to be saved to a database or file somewhere.

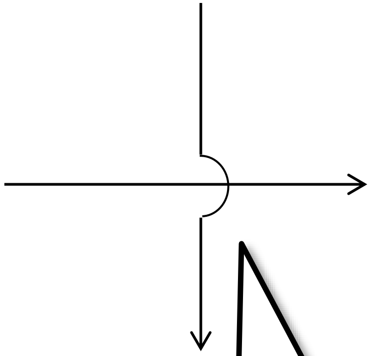
Good & Bad practice

Bad Practice



Lines should never cross like this in a flowchart as it makes following the logic difficult. It can also get quite messy if many lines are crossing.

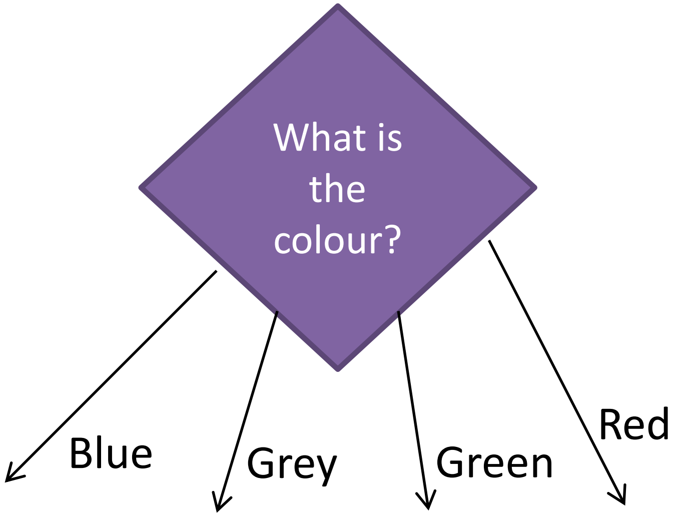
Good Practice



They should always either go around each other OR use bridges like so to indicate which line is which.

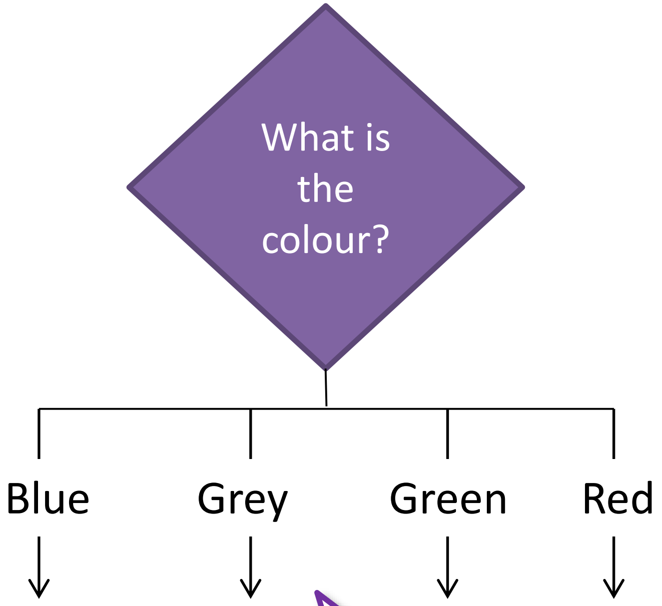
Good & Bad practice

Bad Practice



Decisions *can* have multiple outputs, but displaying them in this hap-hazard way is not correct.

Good Practice



Options should be neatly presented like so with all lines labelled clearly. In Java programming, this is often how we represent switch cases.

Tips for BEST MARKS!

- ✓ Using the more specific shapes we have looked at in this correctly will make your flowchart look more professional. However, be careful not to overload it with senseless symbols!
- ✓ Check carefully that you have not confused the different symbols; especially the on-page and off-page connectors.
- ✓ Using the right symbols is great, but your flowchart must have strong logic and a good presentation.