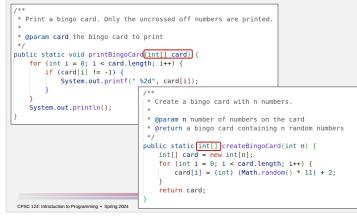
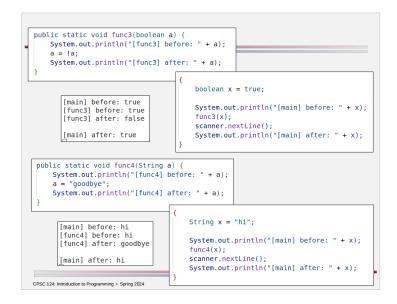
Arrays

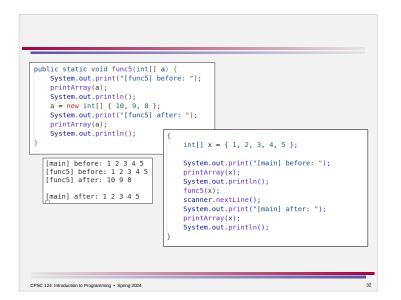
both parameter and return types can be arrays

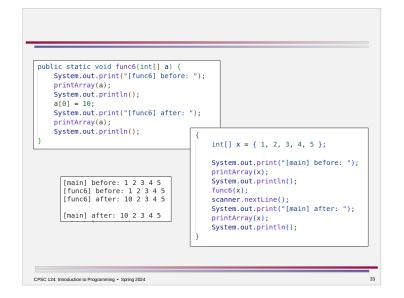




Subtleties of Parameters and Return Values parameters and return values are passed by value a copy of the value of the expression is passed for variables, this is a copy of what is in the box for primitive types (int, double, boolean, etc), the box contains the actual value • for all other types (arrays, String, all types defined by classes), the box contains the address in memory where the actual value is stored this means – assignment to a parameter has only a local effect (the parameter's box is local to the subroutine) assignment to an array slot or invoking a mutable method on an object has a global effect (only the address was copied when the array/object was passed) CPSC 124: Introduction to Programming . Spring 2024







Designing Subroutines

A subroutine's job should be a single complete relatively self-contained task.

- if you can't state the task briefly and without a lot of use of "and", it is probably not a single task
- if the task involves changing many variables used elsewhere, it is probably not a complete or self-contained task
 - return values provide only a limited way for a subroutine to affect its caller
 - the number of parameters is not limited, but there is such a thing as too many

Think of a subroutine as a friend who can do a task for you. If explaining what the task is takes too long, it is too big of a job for one friend. If you have to be too involved in what the friend is doing, you might as well do the task yourself.

Designing Subroutines

In general, a static subroutine should do one (and only one) of the following –

- obtain input
 - print a prompt as needed
 return the input obtained
- produce output
 print rather than return
- compute something
- return the result rather than print
- all necessary values should come via parameters, not input
- change the state of its parameters
- all necessary values should come via
- parameters, not input – no return value or printing

input and output limits a subroutine to particular applications – values can only come from a particular source, specific formatting

subroutines simplify organization

parameters and return values allow the caller to determine where values come from and how they are used

subroutines are reusable

