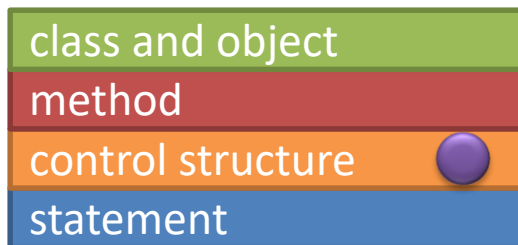


# Selection and Repetition

## Demonstration

Week 5, part 1



- ◼ Making Decisions:
  - Boolean expressions
  - Two-way branching with if and if-else
  - Multi-way branching with switch
  - Enumerated data types

◼ 08 Making Decisions

- ◼ Repeating Actions with Loops, parts 1 & 2

◼ 09 Repeating Actions with Loops



# Tasks starting this week

## 5.1PP Tracing Code with Arrays

- Trace code involving loops and arrays



## 5.2PP Collection of Strings

- Using the notes and provided code, complete a program that processes a collection of words



## 5.3PP Spot the Mistakes

- Apply debugging skills to small pieces of code



## 5.4DN ASCII Paint

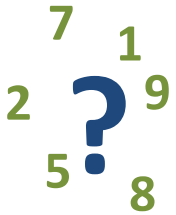
- Work with a 2D array of characters representing a text-based 'image'. Implement paint program functionality 'zoom' and 'fill'





# Number guessing game with do-while

**Task:** user must guess a randomly generated number (cannot opt-out)



**START** Program has generated a 'hidden' number

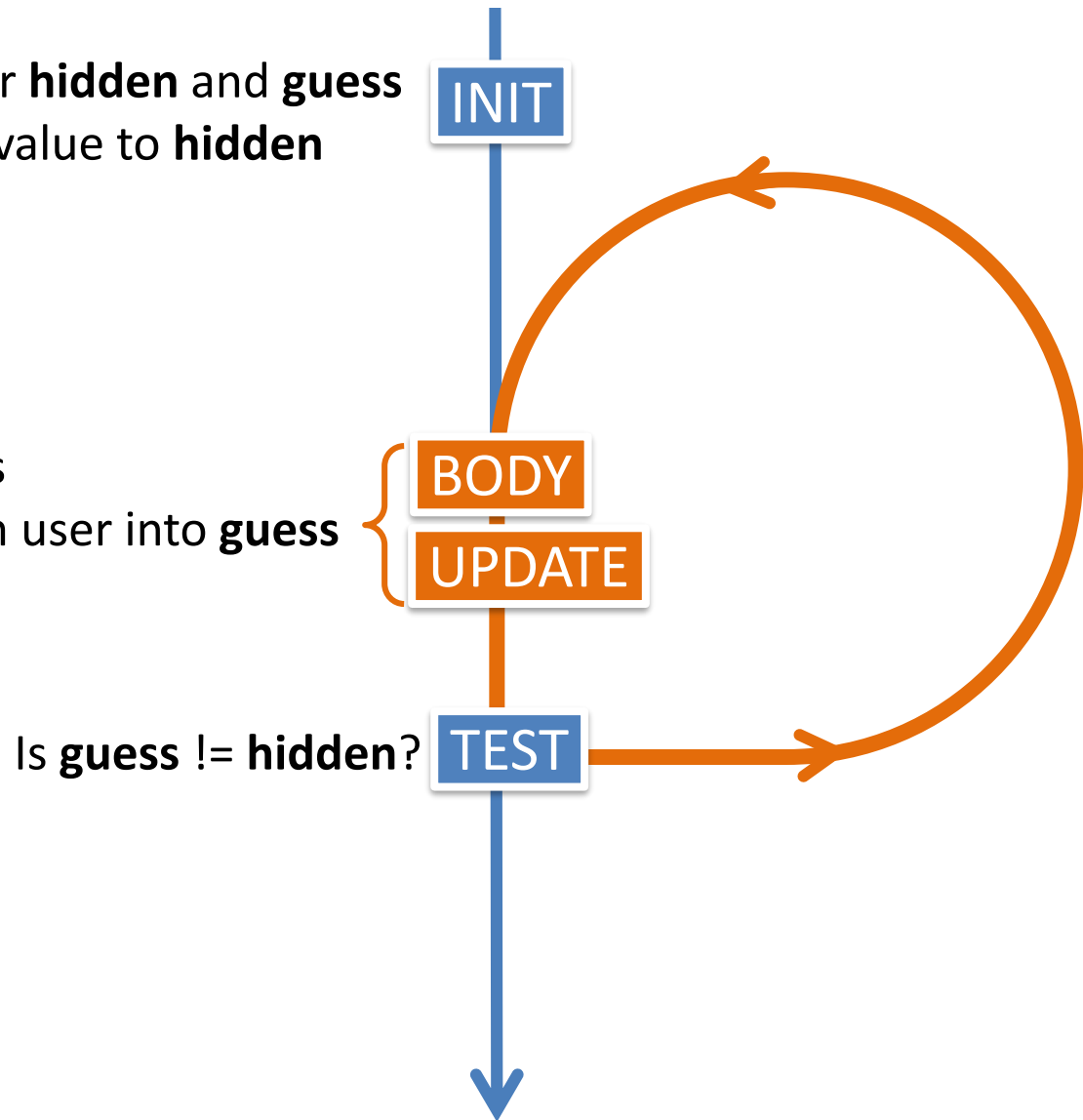
**GOAL** User has found the correct number



# Guessing numbers with do-while

- Allocate integer storage for **hidden** and **guess**
- Assign randomly selected value to **hidden**

- Prompt user for a guess
- Read next number from user into **guess**



# Improved Number Guessing Game

**Task:** An improved guessing game, where  
user may play zero or more times



in each round of the game the program  
randomly selects a secret number between 1  
and 50

user is prompted to guess the number *and*  
for each incorrect guess, given feedback of  
'too high' or 'too low'

on a correct guess the user is congratulated  
and asked if he/she wants to play again

*See `NumberGuessPlus*.java`*



# The problems to solve

START

User wants to play

START

Secret number has been selected

For each game

START

User has entered a guess

For each guess

GOAL

User has been given correct feedback

GOAL

User has guessed secret number

GOAL

User no longer wants to play



# The plan

Revealed during lecture (and in post-lecture slides)

# Introduction to working with arrays

Week 5, part 2

class and object

method

control structure

statement



Arrays



Tracing array code



Methods for working with arrays



Multidimensional arrays (advanced)



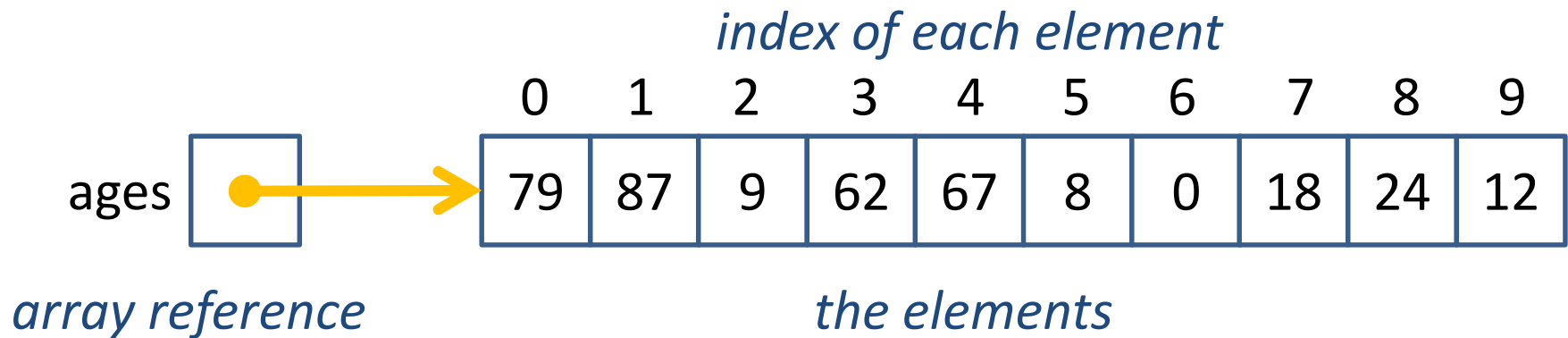
10 Managing Collections with Arrays



# Quick introduction to arrays

An array is an ordered (and indexed) list of values of the same type (primitive or object)

*Example: a list of 10 integer ages*



# Quick introduction to arrays

## Declare an array reference

- syntax: `type[] identifier;`
- example: `int[] a;`

a 

null
------

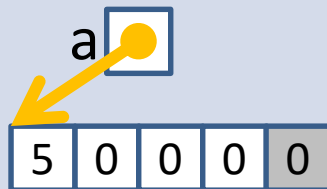
## Allocate space

- syntax: `identifier = new type[size];`
- example: `a = new int[5];`



## Access a specific element

- syntax: `identifier[index];`
- examples:
  - `a[0] = 5;`
  - `int x = a[4];`



## And...

- `array.length` is length of array, as in `a.length`
- Array contents can be modified by methods



# The need for arrays

**Task:** Calculate the average age of a group of people

Possible (partial) solution:

```
int age1, age2, age3, age4;  
int sum;  
double average;  
//Would read ages from user  
//  (~8 lines of code)  
//Would add those together,  
//  storing total in sum  
average = (double) sum / 4;
```



# Improving the (average) Ages application

0. Our starting point

0.5 Replace many *agen* variables with a single array

1. Make use of array indexing by writing repeating code

- to read each age from the user; and
- to calculate the sum

2. Make number of entries more flexible



# Portfolio tasks, next 3 weeks

5.1PP Tracing Code with Arrays

5.2PP Collection of Strings

5.3PP Spot the Mistakes

5.4DN ASCII Paint (2D arrays)

6.1PP Objects as Records

6.2CR Objects with More Abilities

6.3DN Sorting Algorithms

7.1PP Arrays of Objects

7.2PP Structure Charts

5.2: Similar to the Ages example, but with a menu to select options and functions in their own methods

6.1, 6.2: Defining your own data types with multiple properties

7.1: A menu-driven application for managing collections of your custom data type

Custom Program

7.3CR Design



7.4DN Implementation



7.5HD Improved