Array of Structs
Arrays of Structs

- Games of an entire season

- Data Structure
  - Separate variables
  - `game1` and `game2`
  - Not practical
Arrays of Structs

- Games of an entire season

- Data Structure
  - Separate variables
  - `game1` and `game2`
  - Not practical

- Array of structs -- easy implementation
Declaring the array of structs

// constant for the # of games
const int NUMGAMES = 100;

// declare the actual array variable
Game games[NUMGAMES];

one-dimensional array has 100 elements
each element is a Game
the array

games [0]

•
•

games [99]

games
one 'Game'

- season
- datePlayed
- opponent
- location
- W_L_T
- UM_score
- opponent_score
the array

games [0]

games [1]
the array

games [0]

games [1]

games
### Arbitrary "games[i]"

<table>
<thead>
<tr>
<th>games[i].season</th>
<th>games[i].datePlayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>games[i].opponent</td>
<td>games[i].location</td>
</tr>
<tr>
<td>games[i].W_L_T</td>
<td>games[i].UM_score</td>
</tr>
<tr>
<td>games[i].opponent_score</td>
<td>games[i].opponent_score</td>
</tr>
</tbody>
</table>

`games[i]`
Accessing this array

- entire array
Accessing this array

- entire array

  games  type array of Game
Accessing this array

- entire game record at entry $i$
Accessing this array

- entire game record at entry $i$

  games[i]  type Game
Accessing this array

- opponent of game at entry $i$
Accessing this array

- opponent of game at entry $i$

`games[i].opponent` type `string`
Accessing this array

- char in the $j^{th}$ position of the opponent of the game at index $I$
char in the $j^{th}$ position of the opponent of the game at index $i$

games[i].opponent[j]  type char
Rules for passing an ARRAY
- arrays are ALWAYS passed by reference
- use `const` to block changes
- example call:
  ```
  processAllGames ( games );
  ```
array of type: Game
Example prototypes

- no changes allowed to array "games"
  ```c
  void processAllGames(const Game games[]);
  ```

- modifications allowed by the function
  ```c
  void processAllGames(Game games[]);
  ```
Call with an element of array

```
processOneGame ( games[i] );
```

- Passing Single game record,
  - follow rules for passing a STRUCT
  - pass by value or by reference
Pick a Data Structure to hold...

- ...all of the states in the USA
- ...the season statistical totals for a football player
- ...the service information for a car
- ...the daily price of a stock for the last 6 months
- ...the ID numbers of passengers on a plane
- ...the number of stars in each square degree of the sky
- ...the directory information for a University student
- ...the directory information for all University students
Summary

- Review of 'struct'
- Array of struct's
- Declaring
- Accessing
- Passing as parameter
- Prototypes
- Data structures
  - choosing