Today’s Topics

• Section 12
• Abstract Data Types
• `void *`
Section 12

Who has read it?

Any questions?

What I think is important:

ADTs are an abstraction of data storage: separation of interface / behaviour and implementation.

`void *` is an “abstraction” (better: generalization) of data types

Using `void *` shifts the responsibility for type-safety to the programmer!

ADTs + `void *` allows for separation between the algorithm and the data.
#include <assert.h>

int main(void) {
    // data -> void *
    int i = 136;
    void *pV = &i;

    // void * -> data
    int *pI = pV;
    int j = *pI;

    assert(i == j);
}
Section 12 – Separating Algorithm from Data Types

1. `int arr_max(int *arr, int arr_len) {`
2.   assert(arr);
3.   assert(arr_len > 0);
4.   `int max_so_far = arr[0];`
5.   `for (int i = 1; i < arr_len; ++i) {
6.     if (arr->data[i] > max_so_far) {
7.       max_so_far = arr->data[i];
8.     }
9.   }`
10.  `return max_so_far;`
11. }

Blue: part of the algorithm (here: find max in array)
Red: specific to the stored data type (here: int)
Exercise 1 – \texttt{int} \rightarrow \texttt{struct posn}

In \([\texttt{01-int}],\) we have implemented an Array module for storing \texttt{int}. Please have a look at the code and familiarize yourself with it.

In \([\texttt{02-posn}],\) we want you to modify the implementation of the Array module, so that it stores \texttt{struct posn} instead of \texttt{int}.

Think carefully about which parts of the implementation have to be changed, because they \textit{depend on the stored data type}, and which ones can remain unchanged, because they \textit{implement a type-independent algorithm}.
Exercise 2 – \texttt{struct posn} \rightarrow \texttt{struct posn} *

In [03-posnp], we want you to modify the implementation of the Array module, so that it stores \texttt{struct posn} * instead of \texttt{struct posn}.

Think carefully about which parts of the implementation have to be changed, because they \textit{depend on the stored data type}, and which ones can remain unchanged, because they \textit{implement a type-independent algorithm}.

Also, please pay attention to the changed return types in the Array module: instead of passing and returning \texttt{struct posn} directly, all function should now pass and return \texttt{struct posn} * exclusively.
Exercise 3 – \texttt{struct posn} * \rightarrow \texttt{void} *

In [\texttt{04-voidp}], we want you to modify the implementation of the Array module, so that it stores \texttt{void} * instead of \texttt{struct posn} *.

Think carefully about which parts of the implementation have to be changed, because they depend on the stored data type, and which ones can remain unchanged, because they implement a type-independent algorithm.

Remember that you have to write helper functions to “connect” the (untyped) Array ADT with the (strongly typed) data.

Also, please pay attention to the changed signature of all Array module function, the added functions in the Posn module, and the added Pos3d module.