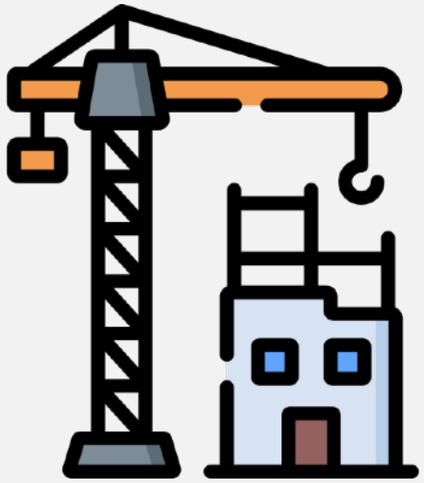
COP 3223H: Introduction to C Programming

Fall 2023

University of Central Florida

Dr. Kevin Moran

Week 13- Class II: Dynamic Structs II



Administrivia

- SPA5 and LPA 3 out today.
- Mid-Semester Feedback Survey will post after class,
  - Will count as Quiz 3 - due Wednesday
- No class on Wednesday, November 22nd - Happy Thanksgiving! 🦃

Today's Agenda

1. Continue Discussion of Dynamic Structs

Review

Indirect Component Selection Operator

- The indirect component selection operator is the character sequence -> placed between a pointer variable and a component name create a reference that follows the pointer to a structure and selects the component.

- While first one is valid to use, it can be a bit cumbersome to use, which is why C provides the indirect operator.

```
book_t *book_ptr = &mybook;

char title[MAX] = (*book_ptr).title;
char title2[MAX] = book_ptr->title;
```

Dynamic Memory and Structs

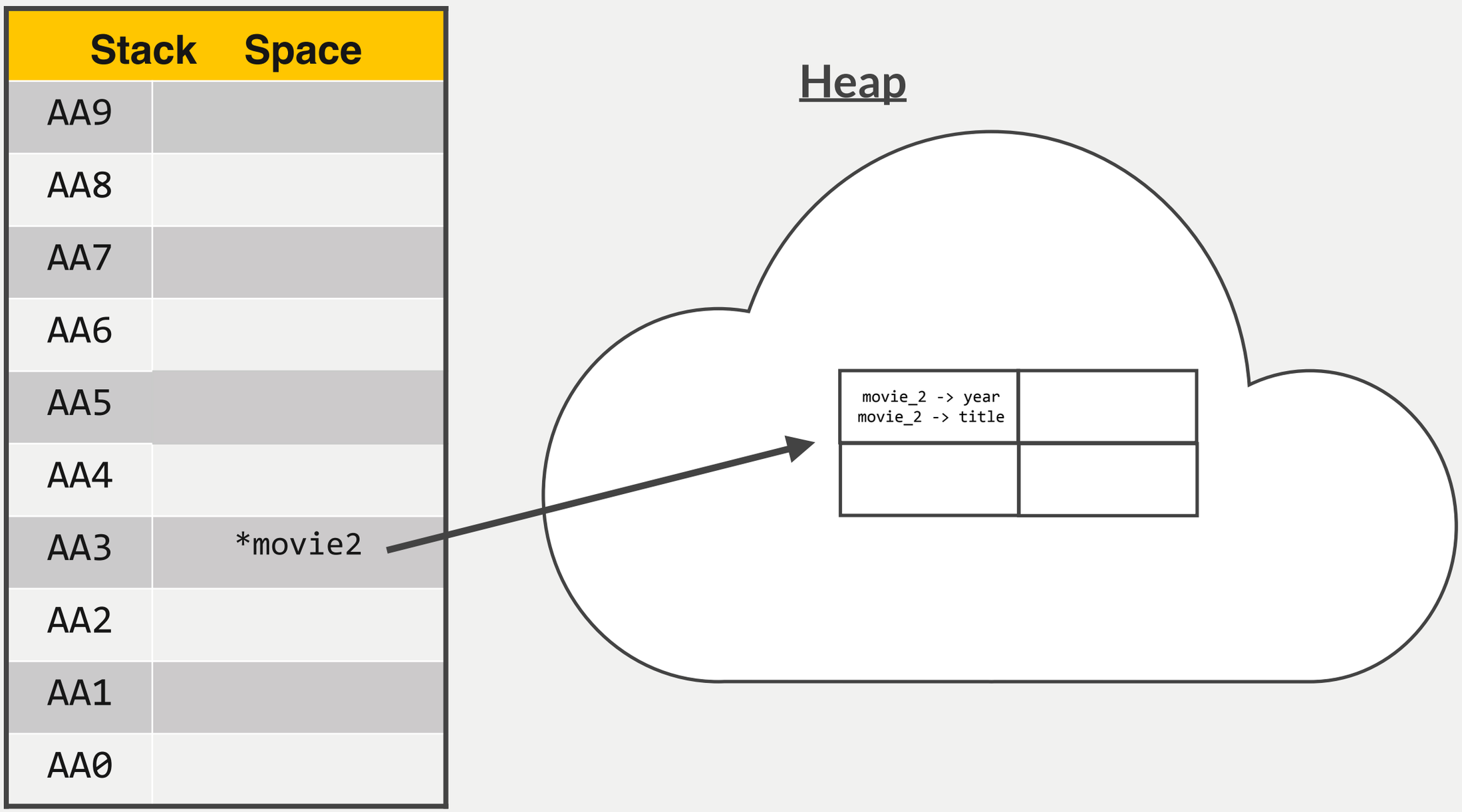
- Similar syntax to dealing with primitive data types.

```
typedef struct{
    int year;
    char title[30];
}movie_t;
```

```
movie_t movie1; // declared in stack space
movie_t *movie2 = (movie_t *) malloc(sizeof(movie_t)); // declared in heap space

free(movie2);
```

Visualizing Dynamic Struct Allocation



Populating the Components of a Dynamic Struct

- Similar syntax to dealing with primitive data types.

```
typedef struct{
    int year;
    char title[30];
}movie_t;
```

```
movie_t *movie2 = (movie_t *) malloc(sizeof(movie_t));

strcpy(movie2->title, "Avatar");
movie2->year = 2022;

printf("%s\n", movie2->title) ;
printf("%d\n", movie2->year);

free (movie2) ;
```

Dynamic Structs (continued)

Dynamic Struct Components

```
typedef struct{
    int year;
    char * title;
    char * author;
}book_t;
```

Dynamic Struct Components

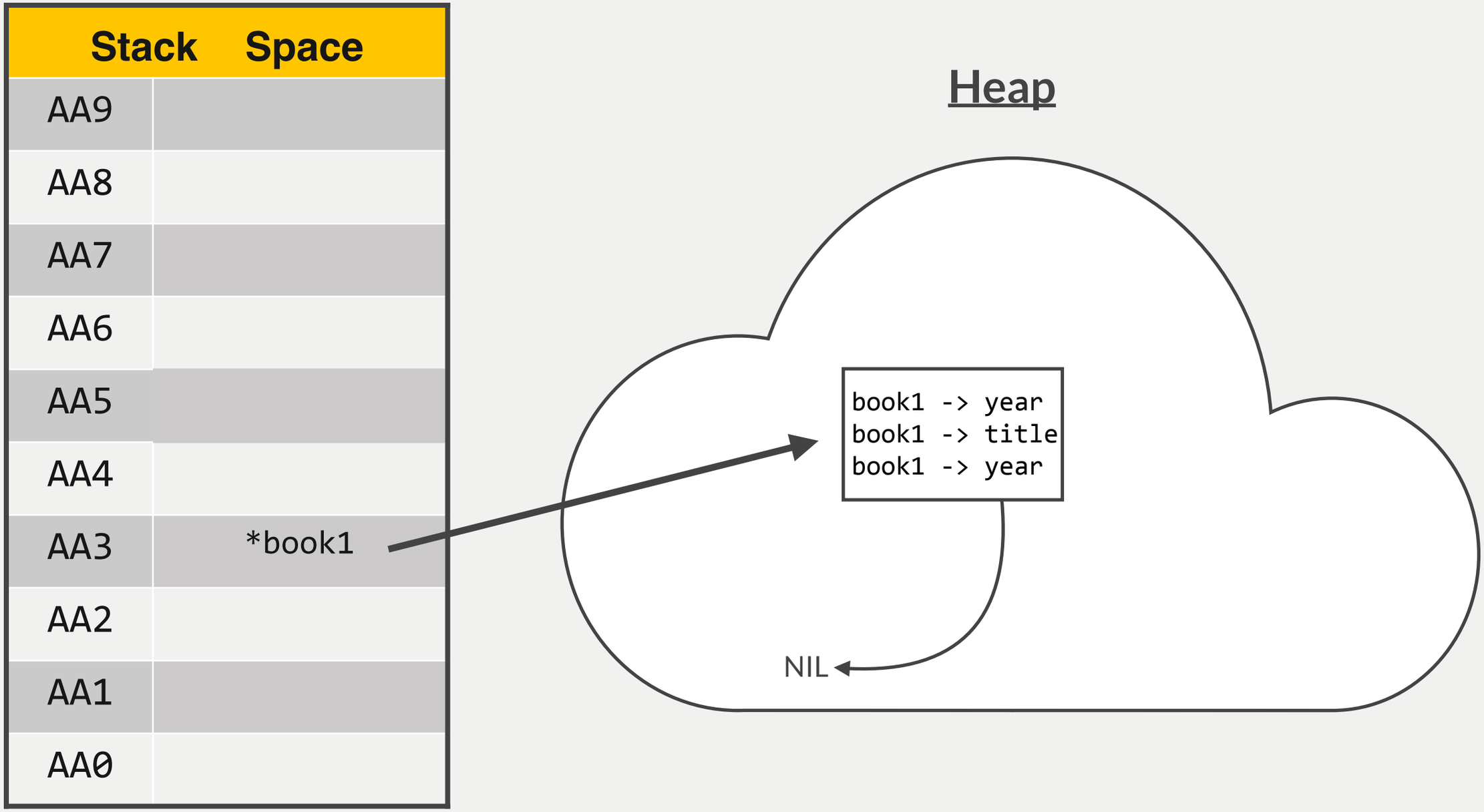
```
typedef struct{
    int year;
    char * title;
    char * author;
}book_t;
```

Anything wrong with this?

```
book_t * book1 = (book_t *) malloc(sizeof(book_t));

strcpy(book1->title, "Harry Potter and the Goblet of Fire");
```

Visualizing Dynamic Struct Allocation



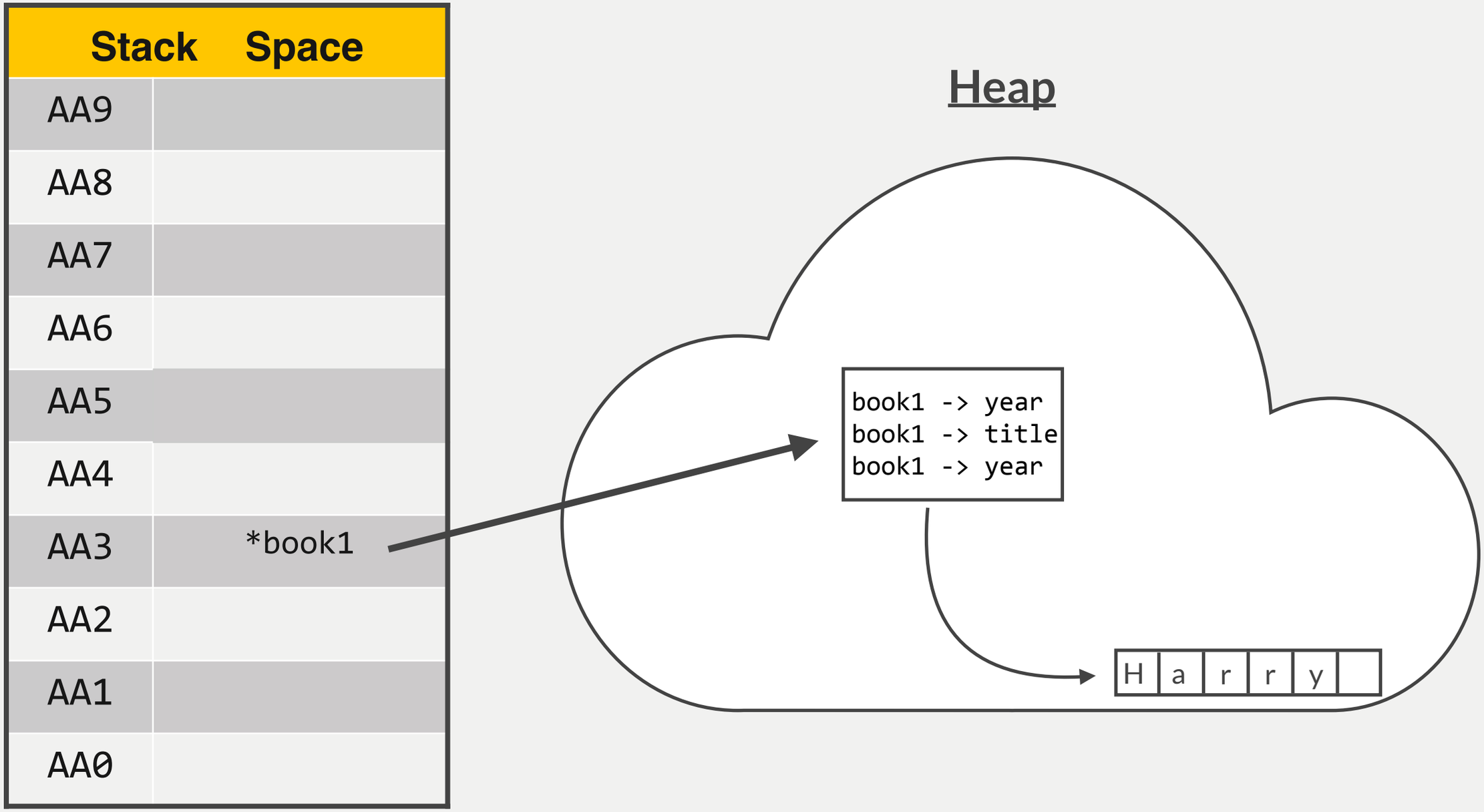
Segmentation Fault!!

Dynamic Struct Components

```
typedef struct{
    int year;
    char * title;
    char * author;
}book_t;
```

```
book_t * book1 = (book_t *) malloc(sizeof(book_t));
book1->title = (char *) malloc(sizeof(char) * 50);
strcpy(book1->title, "Harry Potter and the Goblet of Fire");
```

Visualizing Dynamic Struct Allocation



Deallocating Memory

```
typedef struct{
    int year;
    char * title;
    char * author;
}book_t;
```

Anything
wrong with this?

```
book_t * book1 = (book_t *) malloc(sizeof(book_t));

book1->title = (char *) malloc(sizeof(char) * 50);

strcpy(book1->title, "Harry Potter and the Goblet of Fire”);

book1 -> year = 1998;

free(book1);
```

Deallocating Memory

```
typedef struct{
    int year;
    char * title;
    char * author;
}book_t;
```

Anything wrong with this?

```
book_t * book1 = (book_t *) malloc(sizeof(book_t));

book1->title = (char *) malloc(sizeof(char) * 50);

strcpy(book1->title, "Harry Potter and the Goblet of Fire”);

book1 -> year = 1998;
free(book1->title);
free(book1);
```

Arrays and structs

```
typedef struct{
    int year;
    char * title;
    char * author;
}book_t;
```

- Guess What! We can also have a dynamic array of structs that contains dynamic components!
- The same rules apply that we have been learning with dynamic memory!

```
book_t * mylibrary = (book_t *) malloc(sizeof(book_t) * 10);
```

Acknowledgements

Slides adapted from Dr. Andrew Steinberg's COP 3223H course