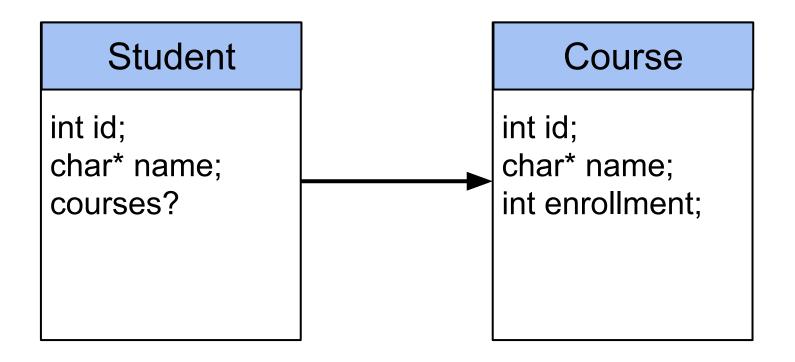
Adapted from materials by Dr. Carrier



Object-oriented programming in C

Object-oriented programming in C (kinda)

Object-oriented programming in C (kinda)

- We can structure our data
 - We're not limited to raw ints, chars, arrays, etc

Structs allow us to group together pieces of data

Structs allow us to group together pieces of data

```
typedef struct Coord {
  double x;
  double y;
} Coord;
```

Structs allow us to group together pieces of data

```
typedef struct Coord {
  double x;
  double y;
} Coord;
```

We can treat Coord like a new type!

Structs allow us to group together pieces of data

```
typedef struct Coord {
  double x;
  double y;
} Coord;
```

We can treat Coord like a new type!

To create and use an *instance* of a struct:

Structs allow us to group together pieces of data

```
typedef struct Coord {
  double x;
  double y;
} Coord;
```

We can treat Coord like a new type!

To create and use an *instance* of a struct:

```
Coord c;
c.x = 5.0;
c.y = 10;
```

- Structs can container raw data types, pointers, arrays, even other structs!

- Structs can container raw data types, pointers, arrays, even other structs!
- We can access members of a struct with .
 - E.g., c.x = 5; double d = c.y;

- Structs still work with pointers!

- Structs still work with pointers!
- We can create a pointer to a struct:

```
Coord*p = &c;
```

- Structs still work with pointers!
- We can create a pointer to a struct:

- We can access members with ->

$$p->x = 1.0;$$

- Structs still work with pointers!
- We can create a pointer to a struct:

- We can access members with ->

$$p->x = 1.0;$$

- This is the same as (*p).x = 1.0;

- Structs still work with pointers!
- We can create a pointer to a struct:

- We can access members with ->

$$p->x = 1.0;$$

- This is the same as (*p).x = 1.0;

We can allocate structs on the stack or the heap!

What are we missing?

What are we missing?

Methods! (member functions)

What are we missing?

Methods! (member functions)

C does not support methods by default.

What are we missing?

Methods! (member functions)

C does not support methods by default.

You can build them, but it's complicated

(function pointers)

Details

```
typedef struct Coord {
  double x;
  double y;
} Coord;
```

```
Coord c;
```

Details

```
typedef struct Coord {
  double x;
  double y;
} Coord;
```

Coord c;

This is only one way to declare a struct.

```
struct Coord2 {
  double x;
  double y;
};
struct Coord2 example;
```

```
typedef defines a new type!

typedef type name;
```

typedef defines a new type!

```
typedef type name;
```

This is not limited to structs:

```
typedef int* int_pointer;
int x = 40;
int_pointer p = &x;
printf("p: %p; *p: %d\n", p, *p);
```

typedef defines a new type!

```
typedef type name;
```

This is not limited to structs:

```
typedef int* int_pointer;
int x = 40;
int_pointer p = &x;
printf("p: %p; *p: %d\n", p, *p);
```

With structs, this prevents us from typing "struct Coord" over and over again!