Why we need structure?
- Each student: ID, name, major, year
- Data structure for each student is compound
  - int ID;
  - char *name;
  - int major;
  - int year;
- 4 pieces of the student?

BAD IDEA

Put Each Piece of People Together
- Like a catcard: A single card contains various information about you
- A structure is a collection of named data items. Each data item is called a field of the structure.

Examples
- struct date {
  char month;
  int day;
  int year;
};

- struct book {
  char *author;
  char *publisher;
  int year;
  int ISBN;
  float price;
};

Structure (cont.)
- Access the fields
  - "." operator
- Memory allocation and alignment
- sizeof()
### Structure Pointer

```c
struct foo {
    int a;
    char b;
    int c;
} n, *np;
np = &n;
```

- `m = np->a`  
- `m = np->b`  
- `m = np->c`

### Structure Operations

```c
struct {
    int x;
    int y;
} a, b, *p;

a = b;
a.x = 2;
p = &a;
foo(a);
return a;
```

- If `a == b` printf("equal"); // No!
- **Then how to compare?**

### Structure as Parameters

- **Pass by value**
  - Copy the whole structure
  - Less efficient
- **Pass by reference**
  - Only pass the pointer to the structure
  - More efficient in memory consumption

### Structure Array

```c
struct key {
    char *word;
    int count;
} keytab[] = {
    {'auto', 0},
    {'break', 0},
    {'case', 0},
    {'char', 0},
    {'const', 0},
};
```

### Data Structures

```
head
```

```
Key
Next
```

```
Key
Next
```

```
Key
Next
```

```
Key
Next
```

```
...NULL
```

```
head
tail
```

```
Key
Next
```

```
Key
Next
```

```
Key
Next
```

```
Key
Next
```

```
...NULL
```
Data Structures (cont.)

typedef: Type Synonym

struct point {
    int x, y;
};
struct point p, *ptr;

typedef struct point POINT;
POINT p, *ptr;

typedef struct {
    int x, y;
} POINT;
POINT p, *ptr;

typedef int WORD;
typedef char BYTE;

Union

union ident {
    int id;
    char name[10];
} n;

All members are stored in the same memory space

struct student {
    int status;
    union ident id;
} n;

Union (cont.)

n.id = 1;
strcpy(n.name, "Gates");

Acknowledgement