

# C

SWE110

Lesson 16

Prof. Zachi Baharav

[zbaharav@cogswell.edu](mailto:zbaharav@cogswell.edu)

# Structures

- Collection of same type --> **Array**
- Collection of various types --> **Structure**

# Example and motivation

```
struct Date{
    int    day;
    int    month;
    int    year;
    char   monthName[80];
};
```

Date : Structure tag (to be used later on)

Different ways to actually create variables of this type:

```
struct Date{
    int    day;
    int    month;
    int    year;
    char   monthName[80];
} today, tomorrow;
```

```
int    ii;
struct Date    today, yesterday;

struct Date    myCalendar[365];
```

# Example and motivation

```
struct Person{
    char first[80];
    char last[80];
    int numRelatives;
    struct Relatives relatives[50];
};
```

```
struct Relatives{
    int relation;
    struct Person* ptr;
};
```

# One more motivation...

- Array of structure:
- Consider a program that counts the occurrence of 'C' keywords.

```
char *keyWord[N_KEY];           // "if" , "for" , "while" , ...
char keyWordCount[N_KEY];      // how many times
```

```
struct KeyWord {
    char *keyWord[80];
    int  keyWordCount;
};
```

```
struct KeyWord keyWordArray[N_KEY];
```

# Accessing elements

```
struct Date today;  
struct Date *ptr;
```

```
today.day = 29;           // . syntax  
today.month = 10;  
ptr = &today;  
ptr->year = 2014;        // -> syntax  
strcpy(ptr->monthName, "October");
```

```
struct Date{  
    int day;  
    int month;  
    int year;  
    char monthName[80];  
};
```

# Traditional

## ("first version of C")

- Only operations allowed with structure are:
  - Access it's member(s).
  - Take the address off.
  - NOT passing to function (as will involve copy of all thing).
  
- RELAXED in newer versions: Can pass by-value structures. Not really recommended...
  - Array-of-structures are passed as arrays, which means just the address of the first element!

# Time to program!!

Dungeon (will be structures)



END