

## Structs

Structs are used for grouping. Sometimes called a *record*, they are the precursor to objects, the cornerstone of C++.

### 1 Defining a Struct

A **struct** is a collection of *members*. There are a couple of ways to set up a struct. One way is to use **typedef**, which is used in general to create a user-defined type. The following example creates a `Date` type which can be used in a program.

```
typedef struct {
    int day;
    int month;
    int year;
} Date;
```

### 2 Accessing a Struct

One has in general direct member access. This uses DOT notation:

```
Date D;
D.day = 1;
D.month = 1;
D.year = 2008;
```

You can have an array of structs. (Indeed you can have an array of anything.) So a program might care about a sequence of events, stored as `Date dates[100]`

### 3 Example Code: `pointing.c`

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int xpos;
    int ypos;
} Point;

const int MAX_PTS=100;
```

```

const int DIM=20;
const char BLANK=' '; const char POINT='X';

int loadPoints(char * fileName,Point list[]); // returns number loaded
void plotPoints(Point list[], int count);

int main(int argc, char * argv[]) {
    Point listing[MAX_PTS];
    int count = loadPoints(argv[1],listing);
    plotPoints( listing, count);
    return 0;
}

// loads points from file
// assumes number of points given at start of file
int loadPoints(char* fileName, Point list[]) {
    FILE* inFile;
    inFile = fopen(fileName, "r");
    if( inFile==NULL ) {
        printf("Couldn't find file %s\n", fileName);
        return 0;
    }

    int p, limit;
    fscanf(inFile, "%d", &limit);
    if(limit>MAX_PTS) {
        printf("Only taking first %d points\n", MAX_PTS);
        limit = MAX_PTS;
    }
    for(p=0; p<limit; p++) {
        fscanf(inFile, "%d%d", &list[p].xpos, &list[p].ypos);
    }

    fclose(inFile);
    return limit;
}

```

```

// plots points on a grid of DIM x DIM
// assumes coordinates in range 0..DIM-1
void plotPoints(Point list[], int count)
{
    char screen[DIM][DIM];
    int i,j,p;
    for(i=0; i<DIM; i++)
        for(j=0; j<DIM; j++)
            screen[i][j] = BLANK;

    for(p=0; p<count; p++) {
        if(list[p].xpos>=0 && list[p].xpos<DIM &&
            list[p].ypos>=00 && list[p].ypos<DIM )
            screen[ list[p].xpos ][ list[p].ypos ] = POINT;
    }

    for(i=0; i<DIM; i++) {
        for(j=0; j<DIM; j++)
            printf("%c", screen[i][j] );
        printf("\n");
    }
}

```