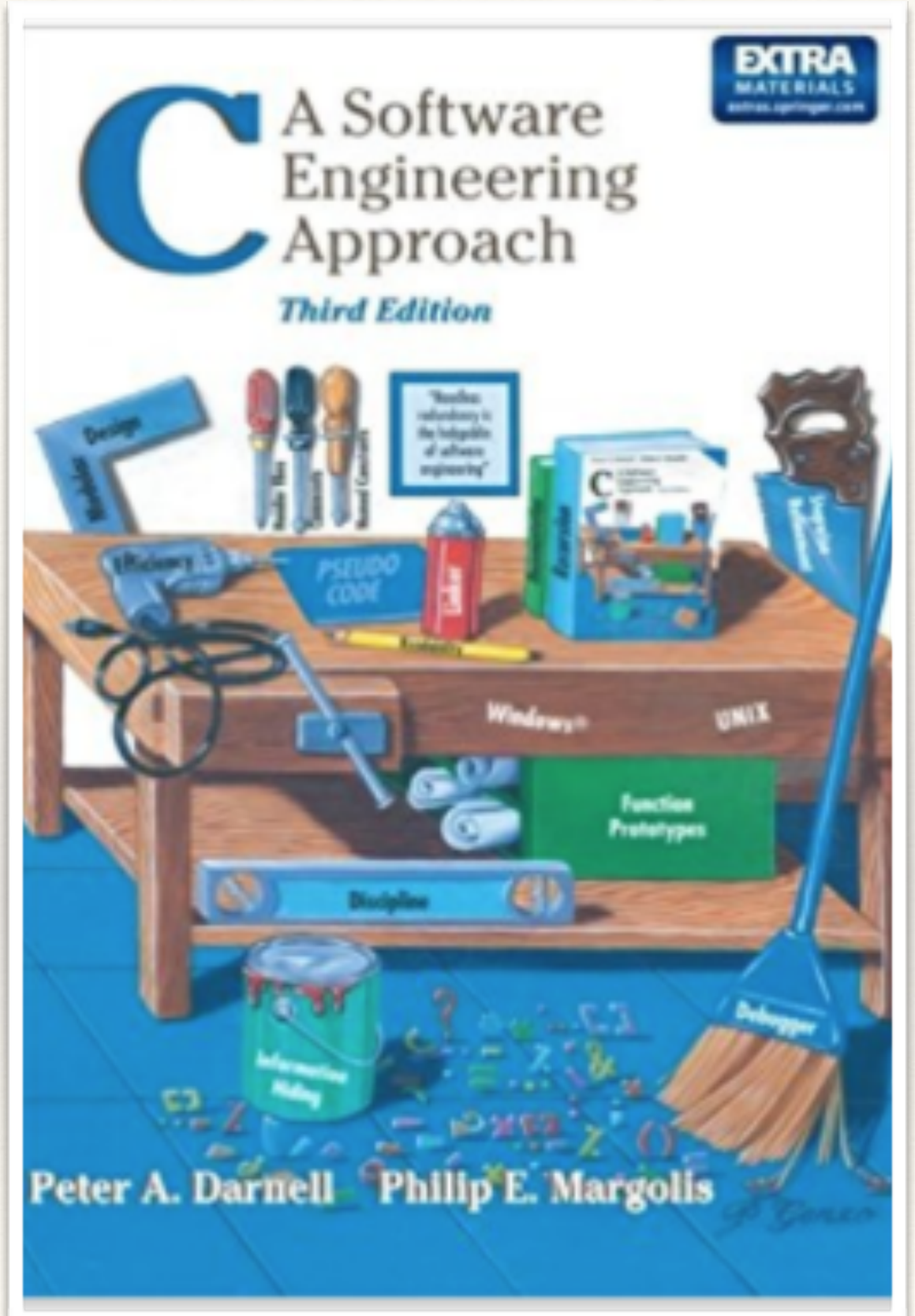
pg 271-278

Unions

Memory is a terrible thing to waste…



What are Unions?

- Unions are similar to structures but…
- Members of a union are overlaid on top of each other so that they share memory.
- You can define a union with many members but only one member can contain a value at any given time.
- Unions allow you to use the same memory for multiple purposes.

Defining a Union

- When you define a union, the compiler allocates enough memory to hold the largest member of the union.
- The data stored depends on which union member is used.

```
union Data {
    char string[20];
    long number;
    float xnumber;
};
Data data;
```

Assignment

```
union Data {
    char string[20];
    long number;
    float xnumber;
};
Data data;


data.number = 10;

data.xnumber = 100.123;

strncpy ( data.string, “Testing”, 8 );
```

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

int main ( )
{
union Data {
   char string[100];
   long number;
   float xnumber;
};
union Data data;

data.number = 10;
printf ( "data: %ld\n", data.number );

data.xnumber = 100.123;
printf ( "data: %f\n", data.xnumber );

strncpy ( data.string, "Testing", 8 );
printf ( "data: %s\n", data.string );

}
```

```
$ ./unionAssign
data: 10
data: 100.123001
data: Testing
```

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

int main ( )
{
union Data {
   char string[100];
   long number;
   float xnumber;
};
union Data data;

data.number = 10;
data.xnumber = 100.123;
strncpy ( data.string, "Testing", 8 );

printf ( "data: %ld\n", data.number );
printf ( "data: %e\n", data.xnumber );
printf ( "data: %s\n", data.string );

}
```

```
$ ./unionAssign1
data: 2911332180553864 4
data: 7.713521e+31
data: Testing
```

```c
#include <stdio.h>
int main ()
{
   union integer {
      unsigned int num[2];
      unsigned long number;
   };

   union integer INTEGER;

   INTEGER.num[0] = 1;
   INTEGER.num[1] = 0;
   printf ( "INTEGER = %d %d\n", INTEGER.num[0], INTEGER.num[1]);
   printf ( "INTEGER.number = %lu\n", INTEGER.number );

   INTEGER.num[0] = 1;
   INTEGER.num[1] = 1;
   printf ( "INTEGER = %d %d\n", INTEGER.num[0], INTEGER.num[1]);
   printf ( "INTEGER.number = %lu\n", INTEGER.number );

   INTEGER.num[0] = 1;
   INTEGER.num[1] = 3;
   printf ( "INTEGER = %d %d\n", INTEGER.num[0], INTEGER.num[1]);
   printf ( "INTEGER.number = %lu\n", INTEGER.number );

}
```

```
$ ./unionBytes
INTEGER = 1 0
INTEGER.number = 1
INTEGER = 1 1
INTEGER.number = 4294967297
INTEGER = 1 3
INTEGER.number = 12884901889
```

0: 1
1: 2
2: 4
3: 8
4: 16
5: 32
6: 64
7: 128
8: 256
9: 512
10: 1024
11: 2048
12: 4096
13: 8192
14: 16384
15: 32768
16: 65536
17: 131072
18: 262144
19: 524288
20: 1048576
21: 2097152
22: 4194304
23: 8388608
24: 16777216
25: 33554432
26: 67108864
27: 134217728
28: 268435456
29: 536870912
30: 1073741824
31: 2147483648
32: 4294967296
33: 8589934592
34: 17179869184
35: 34359738368
36: 68719476736
37: 137438953472
38: 274877906944
39: 549755813888
40: 1099511627776
41: 2199023255552
42: 4398046511104
43: 8796093022208
44: 17592186044416
45: 35184372088832
46: 70368744177664
47: 140737488355328
48: 281474976710656
49: 562949953421312
50: 1125899906842624
51: 2251799813685248
52: 4503599627370496
53: 9007199254740992
54: 18014398509481984
55: 36028797018963968
56: 72057594037927936
57: 144115188075855872
58: 288230376151711744
59: 576460752303423488
60: 1152921504606846976
61: 2305843009213693952
62: 4611686018427387904
63: 9223372036854775808

00000001 **00000000** **00000011** **00000000**

00000000 00000011 00000000 00000001 (**3 1**)

= 8589934592 + 4294967296 + 1

= 12884901889

```c
#include <stdio.h>
#include <string.h>

int main ( ) {

char inputString[50];

struct stats {
   char lastName[50];
   char firstName[50];
   unsigned int citizen;  /* 1 = Canadian */
   union {
      char nationality[50];
      char cityofBirth[50];
   } location;
};

struct stats person;
```

```
printf ( "Enter your last name: " );
scanf ( "%s", inputString );
strncpy ( person.lastName, inputString, strlen(inputString)+1 );

printf ( "Enter your first name: " );
scanf ( "%s", inputString );
strncpy ( person.firstName, inputString, strlen(inputString)+1 );

printf ( "Are you a Canadian citizen?\n   Enter Yes or No: " );
scanf ( "%s", inputString );

if ( !strcmp ( inputString, "Yes" ) ) {
   person.citizen = 1;
   printf ( "Enter your city of birth: " );
   scanf ( "%s", inputString );
   strncpy ( person.location.cityofBirth, inputString,
             strlen(inputString)+1 );
} else {
   person.citizen = 0;
   printf ( "Enter your nationality: " );
   scanf ( "%s", inputString );
   strncpy ( person.location.nationality, inputString,
             strlen(inputString)+1 );
}
```

```
if ( person.citizen == 1 ) {
   printf ( "Last Name: %s\n", person.lastName );
   printf ( "First Name: %s\n", person.firstName );
   printf ( "Nationality: Canadian\n" );
   printf ( "City of Birth: %s\n", person.location.cityofBirth );
} else {
   printf ( "Last Name: %s\n", person.lastName );
   printf ( "First Name: %s\n", person.firstName );
   printf ( "Nationality: %s\n", person.location.nationality );
}

}
```

```
$ ./unionExample
Enter your last name: Stacey
Enter your first name: Deb
Are you a Canadian citizen?
   Enter Yes or No: Yes
Enter your city of birth: Toronto
Last Name: Stacey
First Name: Deb
Nationality: Canadian
City of Birth: Toronto
```

```
$ ./unionExample
Enter your last name: Staszewski
Enter your first name: Jenny
Are you a Canadian citizen?
   Enter Yes or No: No
Enter your nationality: Polish
Last Name: Staszewski
First Name: Jenny
Nationality: Polish
```

```
struct stats person[4];
unsigned int i = 0;
unsigned int j = 0;
unsigned int count = 4;

while ( i < count ) {
   printf ( "Person %d: ", i+1 );
   printf ( "Enter your last name: " );   . . .

   printf ( "Are you a Canadian citizen?  Enter Yes or No: " );
   scanf ( "%s", inputString );
   if ( !strcmp ( inputString, "Yes" ) ) {
      person[i].citizen = 1;
      printf ( "Enter your city of birth: " );
      scanf ( "%s", inputString );
      strncpy ( person[i].location.cityofBirth, inputString,
                strlen(inputString) + 1 );
   } else {
      person[i].citizen = 0;
      printf ( "Enter your nationality: " );
      scanf ( "%s", inputString );
      strncpy ( person[i].location.nationality, inputString,
                strlen(inputString) + 1 );
   }
   i++;
   printf ( "\n" );
}
```

$ ./unionExample2
Person 1: Enter your last name: Stacey
Enter your first name: Deb
Are you a Canadian citizen? Enter Yes or No: Yes
Enter your city of birth: Toronto

Person 2: Enter your last name: Staszewski
Enter your first name: Jenny
Are you a Canadian citizen? Enter Yes or No: No
Enter your nationality: Polish

Person 3: Enter your last name: Gillis
Enter your first name: Dan
Are you a Canadian citizen? Enter Yes or No: Yes
Enter your city of birth: Milton

Person 4: Enter your last name: Driedzic
Enter your first name: Anthony
Are you a Canadian citizen? Enter Yes or No: No
Enter your nationality: Polish

Output:
Person 1: Last Name: Stacey
First Name: Deb
Nationality: Canadian
City of Birth: Toronto

Person 2: Last Name: Staszewski
First Name: Jenny
Nationality: Polish

Person 3: Last Name: Gillis
First Name: Dan
Nationality: Canadian
City of Birth: Milton

Person 4: Last Name: Driedzic
First Name: Anthony
Nationality: Polish

```c
#include <stdio.h>
int main ( ) {

union Data {
   char string[20];
   long number;
   float xnumber;
};

union Data data;

printf ( "sizeof (data.string): %ld\n", sizeof(data.string) );
printf ( "sizeof (data.number): %ld\n", sizeof(data.number) );
printf ( "sizeof (data.xnumber): %ld\n", sizeof(data.xnumber) );

printf ( "sizeof (data): %ld\n", sizeof(data) );

}
```

```c
#include <stdio.h>
int main ( ) {

union Data {
   char string[20];
   long number;
   float xnumber;
};

union Data data;

printf ( "sizeof (data.string): %ld\n", sizeof(data.string) );
printf ( "sizeof (data.number): %ld\n", sizeof(data.number) );
printf ( "sizeof (data.xnumber): %ld\n", sizeof(data.xnumber) );

printf ( "sizeof (data): %ld\n", sizeof(data) );

}
```

```
$ ./unionSize
sizeof (data.string): 20
sizeof (data.number): 8
sizeof (data.xnumber): 4
sizeof (data): 24
```

```c
#include <stdio.h>

int main ( ) {

union Data {
   char string[64];
   long number;
   float xnumber;
};

union Data data;

printf ( "sizeof (data.string): %ld\n", sizeof(data.string) );
printf ( "sizeof (data.number): %ld\n", sizeof(data.number) );
printf ( "sizeof (data.xnumber): %ld\n", sizeof(data.xnumber) );

printf ( "sizeof (data): %ld\n", sizeof(data) );

}
```

```c
#include <stdio.h>

int main ( ) {

union Data {
   char string[64];
   long number;
   float xnumber;
};

union Data data;

printf ( "sizeof (data.string): %ld\n", sizeof(data.string) );
printf ( "sizeof (data.number): %ld\n", sizeof(data.number) );
printf ( "sizeof (data.xnumber): %ld\n", sizeof(data.xnumber) );

printf ( "sizeof (data): %ld\n", sizeof(data) );

}
```

```
$ ./unionSize2
sizeof (data.string): 64
sizeof (data.number): 8
sizeof (data.xnumber): 4
sizeof (data): 64
```

$ getconf LONG_BIT