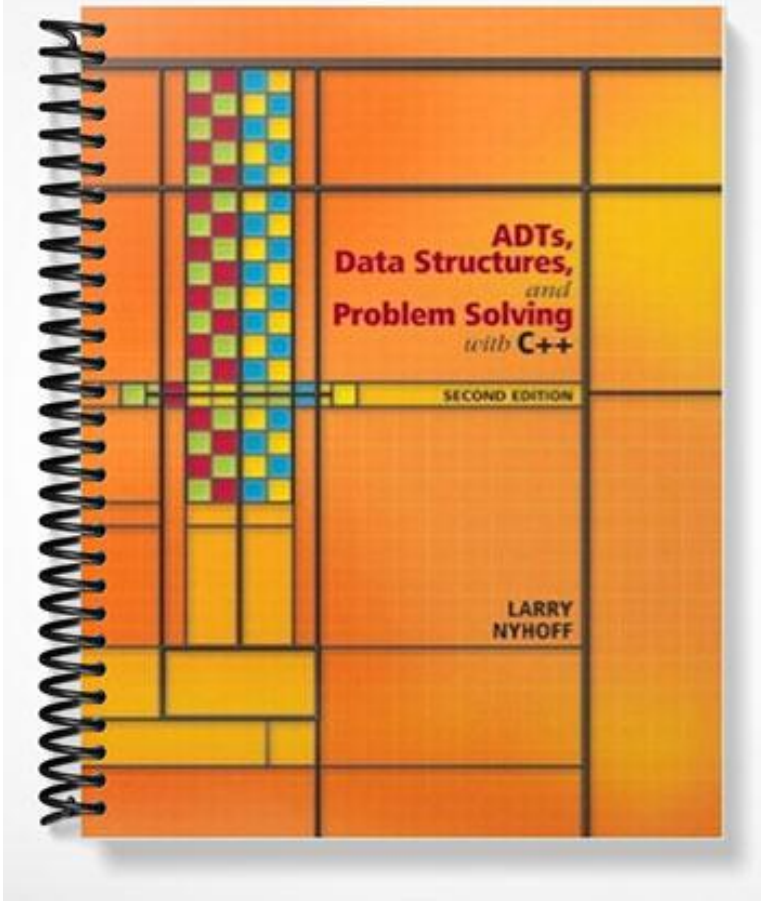
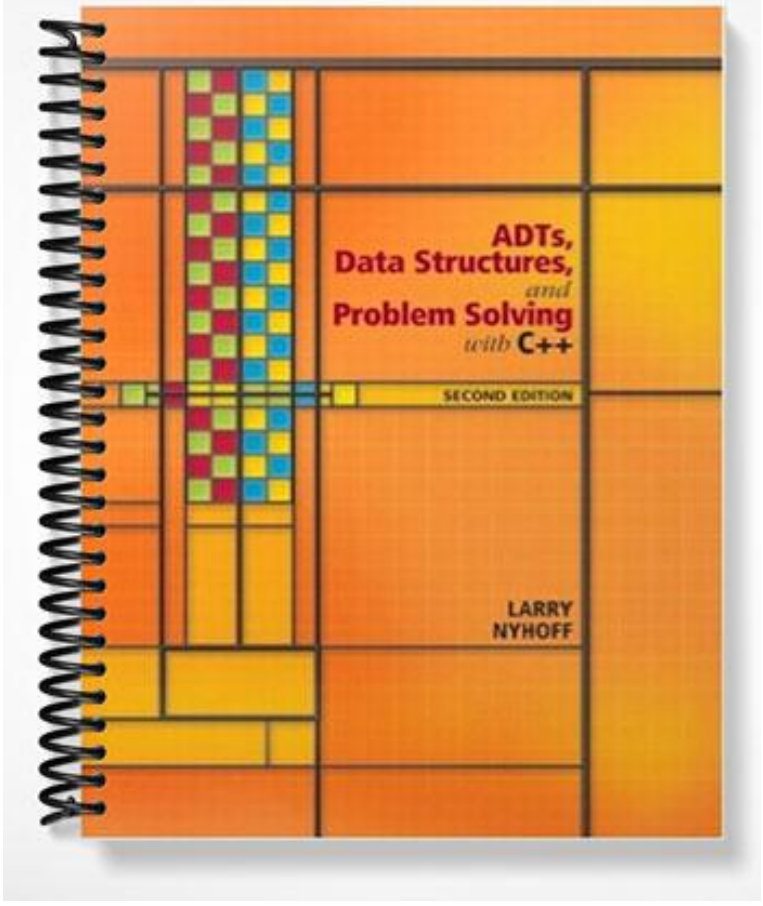
SOLUTIONS MANUAL



SOLUTIONS MANUAL



Chapter Two: ADTs -- C-Style types

Exercises 2.2

1. 0000 0000 0110 0011
2. 0001 0100 1010 0000
3. 0000 0000 1111 1111
4. 1111 1111 0000 0001
5. 0000 0100 0000 0000
6. 1111 1100 0000 0000

7. a) 0011 1111 0010 0000 0000 0000 0000 0000 b) same as (a)
8. a) 0100 0001 1100 1101 0000 0000 0000 0000 b) same as (a)
9. a) 0100 0001 0110 1100 1000 0000 0000 0000 b) same as (a)
10. a) 0011 1100 1000 0000 0000 0000 0000 0000 b) same as (a)
11. a) 0011 1101 1100 1100 1100 1100 1100 1100
    b) 0011 1101 1100 1100 1100 1100 1100 1101
12. a) 0100 0000 0000 0000 1010 0011 1101 0111 b) same as (a)

13. 01000010 01000101
14. 01100010 01100101
15. 01000001 01000010 01001100 01000101
16. 01001110 01101111 00100000 01000111 01101111 00100001
17. 00110001 00110010 00110011 00110100
18. 00110001 00110010 00101110 00110011 00110100

19. 64
20. 28271
21. –16386
22. –16383
23. –26215
24. –21846

25. 3.0
26. 6.0
27. 504.0
28. 1.5
29. 213×2^{36}
30. –3.375

31. null @
32. no
33. false true
34. true true

Exercises 2.3

1. `typedef char byte;`
2.
```
typedef float Real;
typedef float SinglePrecision;
```
3.
```
enum MonthAbbrev {JAN, FEB, MAR, APR, MAY, JUN,
                  JUL, AUG, SEP, OCT, NOV, DEC};
```
4. true
5. true
6. `OCT`
7. `MAR`
8. `OCT`
9. `JUN`
10.
```
enum Digit {ZERO, ONE, TWO, THREE, FOUR,
            FIVE, SIX, SEVEN, EIGHT, NINE};
```
11.
```
enum CURRENCY {PENNY = 1, NICKEL = 5, DIME = 10,
               QUARTER = 25, HALF_DOLLAR = 50, DOLLAR = 100};
```

Exercises 2.4

1. 1. Specifies how memory is to be allocated for that variable
   2. Associates the variable's name with that memory
   3. Initializes that memory with values provided in the declaration (if any).

2.
```
double * p1, * p2;
```

3.
```
p1 = &d1;
p2 = &d2;
```

4. Not possible – `i2` is an integer, but `p1` can only store addresses of `double`s.

5.
```
int * ptr1 = &i1,
    * ptr2 = &i2;
```

6.
```
p1 = p2;
```

7.
```
*ptr1 = *ptr2;
```

8.
```
double temp = *p1;
*p1 = *p2;
*p2 = temp;
```

9.
```
typedef char * CharPointer;
```

10. Machine-dependent results (typically 4)

11. Machine-dependent results (typically 4)

12. Machine-dependent results (typically 8)

13. Machine-dependent results (typically 2)

14. Machine-dependent results (typically 4)

15. Machine-dependent results (typically 4)

16.
```
1
13
43
55
77
99
```