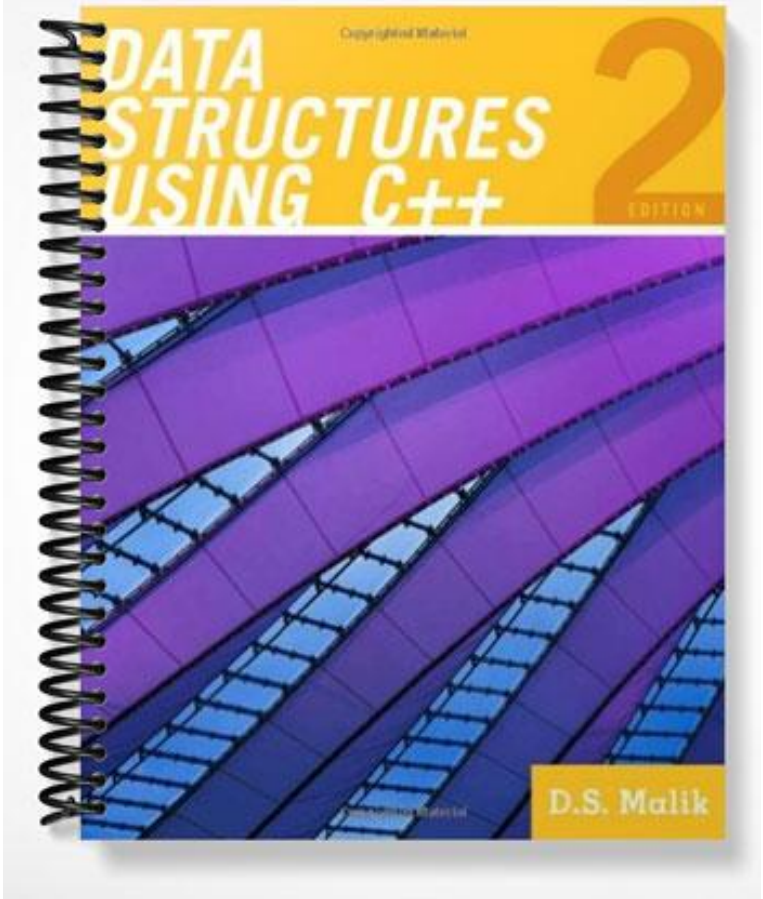
SOLUTIONS MANUAL



ANSWERS TO EXERCISES

Chapter 1

1. a. true; b. false; c. false; d. false; e. false; f. true; g. false; h. false

2. The black-box refers to testing the correctness of the program; that is, making sure that the program does what it is supposed to do. In black-box testing, you do not know the internal working of the algorithm or function. You know only what the function does. Black-box testing is based on inputs and outputs.

3. The white-box refers to testing the correctness of the program; that is, making sure that the program does what it is supposed to do. White-box testing relies on the internal structure and implementation of a function or algorithm. The objective is to ensure that every part of the function or algorithm is executed at least once.

4. Precondition: The value of `x` must be nonnegative.
   Postcondition: If the value of `x` is nonnegative, the function returns the positive square root of `x`; otherwise, the program terminates.

5. a. $O(n^2)$
   b. $O(n^3)$
   c. $O(n^3)$
   d. $O(n)$
   e. $O(n)$
   f. $O(n\log_2 n)$

6. 12

7. a. 43
   b. $4n + 3$
   c. $O(n)$

8. -51, -50, -49, -1, 0, 1, 49, 50, 51

9. One possible answer is as follows:

```
int sumSquares(int n)
{
       int sum = 0;

       for (int j = 1; j <= n; j++)
           sum = sum + j * j;

       return sum;
}
```

   The function `sumSquares` is of the order $O(n)$.

10. The `for` loop has n iterations. Each time through the loop a fixed number of statements execute. Hence, this algorithm is $O(n)$. Now each time through the loop there are two additions . Thus, the number of additions is $2n$.
11. The `for` loop has $2n$-4 iterations. Each time through the loop a fixed number of statements execute. Hence, this algorithm is $O(n)$. Now each time through the loop there is one addition, one subtraction, and one multiplication. Thus, the numbers of additions is $2n$-4, the number of subtractions is $2n$-4, and the number of multiplications is $2n$-4.

12. The outer `for` loop has $2n$ iterations. For each iteration of the outer loop, the inner loop has n iterations. Hence, the total number of iterations of these loops is $2n \times n = 2n^2$. This implies that this algorithm is $O(n^2)$.
13. There are three nested `for` loop and each of these loops has n iterations. For each iteration of the outer loop, the middle loop has n iterations. Thus, the middle loop executes n times and has n^2 iterations. For each iteration of the middle loop, the inner most loop has n iterations. It follows that the inner most loop has n^3 iterations. Hence, this algorithm is $O(n^3)$.
14. a. Constructors have no type. Therefore, the statement:

```
int AA(int, int);
```

should be :

```
AA(int, int);
```

b. Missing semicolon after `}`.
c. There should be a `:` after the member access specifier `public`. (Replace `;` with `:` after the label `public`.)

15.
a. 6
b. 2
c. 2
d.

```
void xClass::func()
{
    u = 10;  v = 15.3;
}
```

e.

```
void xClass::print()
{
    cout << u << " " << v << endl;
}
```

f.

```
xClass::xClass()
{
    u = 0;
    v = 0;
}
```

g. `x.print();`
h. `xClass t(20, 35.0);`

16.
a. (i) Constructor at Line 1
(ii) Constructor at Line 3
(iii) Constructor at Line 4
b.

```
CC::CC()
{
    u = 0;
    v = 0;
}
```

c.

```
CC::CC(int x)
```

```
{
    u = x;
    v = 0;
}
```

d.

```
CC::CC(int x, int y)
{
    u = x;
    v = y;
}

CC::CC(double x, int y)
{
    u = y;
    v = x;
}
```

17.
```
00:00:00
23:13:00
06:59:39
07:00:39
The two times are different.
```

18. (a)-(c)

```
class secretType
{
public:
    void print() const;
    void setName(string);
    void setAge(int);
    void setWeight(int);
    void setHeight(double);
    string getName() const;
    int getAge() const;
    int getWeight() const;
    int getHeight() const;
    secretType(string = "", int = 0, int = 0, double = 0.0);

private:
    string name;
    int age;
    int weight;
    double height;
};
```

d.

```
void secretType:: print() const
{
    cout << "Name: " << name << endl;
    cout << "Age: " << age << endl;
    cout << "Weight: " << weight << endl;
    cout << "Height: " << height << endl;
}

void secretType::setName(string n)
{
    name = n;
}
```

```
void secretType::setAge(int a)
{
    age = a;
}

void secretType::setWeight(int w)
{
    weight = w;
}

void secretType::setHeight(double h)
{
    height = h;
}

string secretType::getName() const
{
    return name;
}

int secretType::getAge() const
{
    return age;
}

int secretType::getWeight() const
{
    return weight;
}

int secretType::getHeight() const
{
    return height;
}

secretType::secretType(string n, int a, int w, double h)
{
    name = n;
    age = a;
    weight = w;
    height = h;
}
```

19. a. `personType student("Buddy", "Arora");`
    b. `student.print();`
    c. `student.setName("Susan", "Miller");`