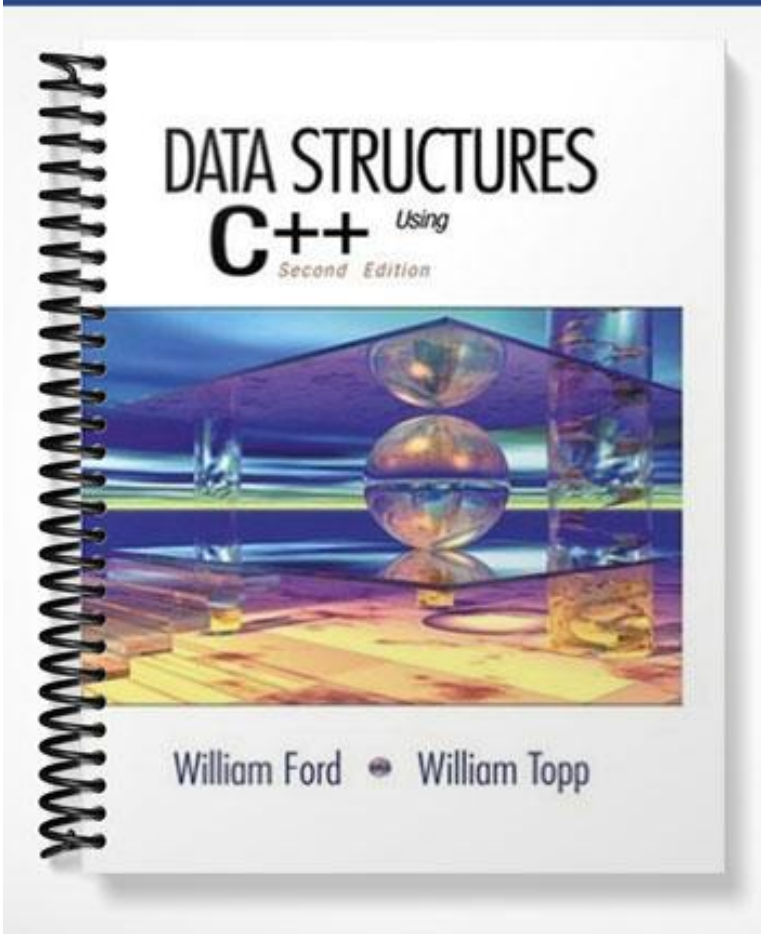
SOLUTIONS MANUAL



Chapter 2
Object Design Techniques

Sample Test

1. The mnemonics OOA, OOD, and OOP describe three phases in the program development life cycle. For each activity, indicate the phase.
   (a) Set up the data structures and their ADT.
   (b) Check whether the proposed software system would be cost effective.
   (c) Perform whitebox testing.
   (d) Setup the functional specifications for the problem.
   (e) Develop the private member functions that will support implementation of a class.
   (f) Determine the public interfaces for the classes and how the classes interact.
   (g) Use the functional specification of the software system to develop an abstract model of the system.

2. A _____ is a computer professional who serves as an intermediary between the client and the software engineer
   (a) Design analyst (b) Systems analyst (c) Client consultant (d) Software engineering designer

3. The calendar class uses a default starting date of
   (a) January, 1980 (b) January, 2000 (c) January, 1900 (d) January, 0000

4. Suppose that we know that the day corresponding to January 1 of the current year is 3 (Wednesday). Describe the algorithm that determines the day corresponding to August 1 of the current year.

5. Verifying that we are solving the correct problem is called _____; testing that we are solving the problem correctly is called _____

6. _____ testing focuses on the input and output behavior of a program without concern for the internal structure of its classes and functions

7. _____ testing focuses on the internal structure of the program

8. The function slideBack () scans elements in an array starting at the first element. The function looks at successive pairs of elements and exchanges their values if the first element is greater than the second element. At the end of the scan, the largest element should appear at the back (end) of the array. We provide the following incorrect implementation for slideBack().

```
void slideBack(int arr[], int n)
{  int i, temp;

   for (i = 0; i < n; i++)
      if (arr[i] > arr[i+1])
      {  temp = arr[i+1];
         arr[i+1] = arr[i];
         arr[i] = temp;
      }
}
```

What is the error in the implementation of the function?

9. The function contains() scans an array to determine if an item is in the list. The function returns true if the item is found and false otherwise. The following implementation works but is not a good solution.

```
bool contains(int arr[], int n, int item)
{
   int i;
   bool itemFound = false;

   for (i = 0; i < n; i++)
      if (arr[i] == item)
         itemFound = true;

   return itemFound;
}
```

   (a) How might whitebox testing reveal inefficiencies in the code
   (b) What correction would make the code more efficient?

10. The object-oriented programming design model views a software system as a set of objects that interact to carry out a task. The top-down design model views the system as a hierarchical set of subprograms that decomposes the problem into separate components. Discuss how these two models might be used in a program design .

11. Good and bad input for a program often fall within ranges. Data that lie between these ranges are called _____ values and represent values that are one step away from changing program behavior.

12. When a program identifies an input error, good design dictates that the programmer should immediately terminate execution with an "exit()" statement so that the user does not obtain erroneous output.
   (a) True (b) False

13. An exception is a(n) _____ that responds to a program error.
   (a) Boolean flag (b) object (c) constant data value (d) system supplied library function

14. Assume function f() might throw an exception. In order to handle the exception, a programmer must include the call to f() in a _____; the program handles the exception in a code segment called a _____.

15. (a) The statements in a try block may generate only one exception.
      (i) True (ii) False

   (b) There must be a one to one correspondence between a try block and a catch block in the sense that each try block may include only one catch block to handle an exception.
      (i) True (ii) False

16. In the employee class, the member function setPayRate() throws an employeeError exception with the message "Below minimum wage". Assume the constant real number MINPAYRATE defines the current minimum wage.

```
class employee
{
   public:
      . . .
      void setPayRate(double pay);
   private:
      . . .
      double payRate;
};
```

(a) Complete the implementation of the function setPayRate()

```
void employee::setPayRate(double pay)
{ }
```

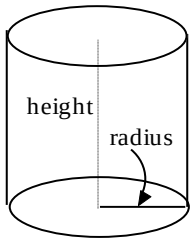
(b) Assume empObj is an object of type employee. Fill in the missing statements in a code sequence that calls the member function and outputs the error message if an exception occurs.

```
_________________
{
    empObj.setPayRate(6.50);
}
_____________________________________
{
    // Output the message
    _____________________________________
}
```

17. What C++ object technology concept is involved when a class has a data member which is itself an object of class type. _____

18. With object composition, a class that is included by composition is called the _____ class and the class that includes an object by composition is called the _____ class.

19. The cylinder class provides measurement of a cylinder object, which we can view as a 3-dimensional figure created by a circle sliding vertical upward along a line denoting the height. A cylinder is defined by the radius of the circular base and the height. Using object composition, we represent the base using a circle object.




```
class cylinder
{
   public:
      cylinder(double r, double h): _______, height(h)
      {}

      double volume()          // area of base * height
      { _________________}

      double getRadius()       // radius of the base
      { return _________ }

   private:
      circle base;
      double height;
};
```

The following is the circle class implementation using inline code.

```
const double PI = 3.141592653589793;

   class circle
   {
      public:
         circle(double r = 0.0): radius(r)
         {}

         double getRadius() const
         { return radius; }

         void setRadius(double r)
         { radius = r; }

         double area() const
         { return PI * radius * radius; }

         double circumference() const
         { return 2.0 * PI * radius; }

      private:
         double radius;
      };
```

(a) In the implementation of the cylinder constructor, complete the initialization list.
(b) Give the return value for getRadius().
(c) Implement the member function volume().

20. Companies hire "temp" workers at an hourly rate and pay them at the end of the day. The tempWorker class uses the composition of two time24 objects to indicate when the worker begins work and ends work. The third data member is the hourly pay rate.

```
   class tempWorker
   {
      public:
         // constructor has arguments to initialize both time24 data
         // values and the rate per hour
         tempWorker(double rate, int sthour, int stminute,
                    int endhour, int endminute);

         // return pay for the day using time24 data and rate per hour
         double pay() const;
      private:
         double ratePerHour;
         time24 startwork, endwork;
   };
```

(a) Implement the constructor using an initialization list.
(b) Implement the function pay().
(c) Declare the tempWorker object obj, where the hourly pay rate is $8.50 and the worker begins at 8:30 and ends at 16:15 (4:15 PM).

21. Functions can be overloaded provided they have _____

22. The class testCL overloads the += operator.

```
   class testCL
{
   public:
      . . .
      // overloaded compound addition operator
      testCL& operator+= (const testCL& obj);

   private:
      . . .
};
```

Assume that a and b are objects of type testCL. The operation b += a is implemented as

(a) `b = a.operator+=(b)` (b) `b.a.operator+=()`
(c) `b.operator+=(a)` (d) `b = b.operator+=(a)`

23. The following is the declaration of the operator +, which is overloaded as a friend function in the time24 class.

```
  friend time24 operator+ (const time24& lhs, const time24& rhs);
```

(a) Modify the declaration so that the operator becomes a member function of the class.
(b) Implement the member function using inline code.

24. The class length stores the linear measure of an object with integer data members feet and inch.

```
   class length;
   {
      public:
         // constructor initializes feet and inches
         length(int ft = 0, int in = 0);

         int getFeet() const;
            // return the feet value for the current length

         int getInch() const;
            // return the inch value for the current length

         double getLength() const;
            // return the length as a real number in units of feet

         friend bool operator== (const length& lhs, const length& rhs);
         friend bool operator<  (const length& lhs, const length& rhs);
            // compare the total length of the two operands

         friend length operator+ (const length& lhs, const length& rhs);
            // form and return lhs + rhs; store in standard form

         friend length operator- (const length& lhs, const length& rhs);
            // form and return lhs - rhs; store in standard form
            // Precondition: lhs >= rhs. if not, throw rangeError

         length& operator+= (const length& rhs);
            // current object = object + rhs; store in standard form
            // Postcondition: the length increases by the value of rhs
```

```
      friend ostream& operator<< (ostream& ostr, const length& obj);
         // output obj in the format ft'in"
   private:
      int feet;
      int inch;

      // utility function sets inches in range from 0 to 11
      void standardLength();
};
```

(a) Implement operator<.

(b) In the implementation of operator-, which of the following statements throws a rangeError exception.

(i) `rangeError = exception("length: invalid subtraction");`
(ii) `throw rangeError = exception("length: invalid subtraction");`
(iii) `throw exception rangeError("length: invalid subtraction");`
(iv) `throw rangeError("length: invalid subtraction");`

(c) In the declaration of the stream output operator, why is the ostream argument ostr not declared using constant reference?

(d) Implement operator+=.

(e) Implement member function getLength().

(f) Give the output that results from the (cout) statements in the following code.

```
length lenA(2,3), lenB(4,6), lenC(8,10), lenD;

cout << lenA.getLength();  // output: _____________________
lenA += lenC;
cout << lenA.getFeet();    // output: _____________________
lenD = lenB + lenC;
cout << lenD;              // output: _____________________
```

Test Solutions

1. (a) OOD (b) OOA (c) OOP (d) OOA (e) OOP (f) OOD (g) OOD

2. (b)

3. (c)

4. January 1 of 2003 is a Wednesday. Using this year as an example, the following code determines the integer value of the first day of August.

```
Date d(8,1,2003);
int n =  d.numberOfDays();
int firstDay = 3;
int firstAugust = (firstDay + n)%7;
```

5. program validation program verification

6. Blackbox testing

7. Whitebox testing

8. The for loop should be

```
for (i=0;i < n-1;i++)
```

9. (a) Use a profiler or a debugger to count the number of iterations. Note all calls to the function require n iterations independent of item.

   (b) Replace the if statement by

```
if (arr[i] == item)
   return true;
```

10. Often the design of a public member function partitions the operation into a series of tasks that are performed by private member functions. This partitioning process employs a top-down model.

11. Boundary values

12. (b)

13. (b)

14. try block catch block

15. (a) (ii) (b) (ii)

16. (a)
```
void employee::setPayRate(double pay)
{
   if (pay < MINPAYRATE)
      throw employeeError("Below minimum wage");

   payRate = pay;
}
```

(b)
```
try
{
   empObj.setPayRate(6.50);
}

catch (const employeeError& ee)
{
   // Output the message
   cout << ee.what() << endl;
}
```

17. Object composition

18. supplier class client class

19. (a)
```
cylinder(double r, double h): base(r), height(h)
{}
```

(b)
```
base.getRadius()
```

(c)
```
double volume()
{ return base.area() * height; }
```

20. Companies hire "temp" workers at an hourly rate and pay them at the end of the day. The tempWorker class uses the composition of two time24 objects to indicate when the worker begins work and ends work. The third data member is the hourly pay rate.

```
class tempWorker
{
   public:
      // constructor has arguments to initialize both time24 data
      // values and the rate per hour
      tempWorker(double rate, int sthour, int stminute,
                 int endhour, int endminute);

      // return pay for the day using time24 data and rate per hour
      double pay() const;
   private:
      double ratePerHour;
      time24 startwork, endwork;
};
```

(a)
```
tempWorker::tempWorker(double rate, int sthour, int stminute,
           int endhour, int endminute): ratePerHour(rate),
           startwork(sthour,stminute), endwork(endhour,endminute)
{}
```

(b)
```
double tempWorker::pay()const
{
   time24 workTime = endwork - startwork;

   return ratePerHour*(workTime.getHour() + workTime.getMinute()/60.0);
}
```

(c)
```
tempWorker obj(8.50, 8, 30, 16, 15);
```

21. Functions can be overloaded provided they have distinct argument lists.

22. (c)

23. The following is the declaration of the operator +, which is overloaded as a friend function in the time24 class.

```
friend time24 operator+ (const time24& lhs, const time24& rhs);
```

(a)
```
time24 operator+ (const time24& rhs);
```

(b)
```
time24 operator+ (const time24& rhs)
{
   return time24(hour+rhs.hour, minute+rhs.minute);
}
```

24. (a)
```
bool operator< (const length& lhs, const length& rhs)
{
   return (lhs.feet*12 + lhs.inch) < (rhs.feet*12 + rhs.inch);
}
```

(b) (iv)

(c) All streams have state changes during an operation and can never be const.

(d)
```
length& length::operator+= (const length& rhs)
{
   *this = *this + rhs;

   return *this;
}
```

(e)
```
double length::getLength() const
{
   return feet + inch/12.0;
}
```

(f)
```
output: 2.25
output: 11
output: 13' 4"
```