CHAPTER 6 REVIEW / STUDY QUESTIONS

Review Questions and Exercises

Fill-in-the-Blank and Short Answer

1. The __________ is the part of a function definition that shows the function name, return type, and parameter list.
2. If a function doesn't return a value, the word __________ will appear as its return type.
3. If function showValue has the following header: void showValue(int quantity) you would use the statement showValue to call it with the argument 5.
4. Either a function's __________ or its __________ must precede all calls to the function.
5. Values that are sent into a function are called arguements.
6. Special variables that hold copies of function arguments are called __________.
7. When only a copy of an argument is passed to a function, it is said to be passed by value.
8. A(n)__________ eliminates the need to place a function definition before all calls to the function.
9. A(n)__________ variable is defined inside a function and is not accessible outside the function.
10. __________ variables are defined outside all functions and are accessible to any function within their scope.
11. __________ variables provide an easy way to share large amounts of data among all the functions in a program.
12. Unless you explicitly initialize numeric global variables, they are automatically initialized to __________.
13. If a function has a local variable with the same name as a global variable, only the __________ variable can be seen by the function.
14. __________ local variables retain their value between function calls.
15. The __________ statement causes a function to end immediately.
16. Default arguments are passed to parameters automatically if no argument is provided in the function call.
17. When a function uses a mixture of parameters with and without default arguments, the parameters with default arguments must be defined __________.
18. The value of a default argument must be a(n)__________.
19. When used as parameters, reference variables allow a function to access the parameter's original argument.
20. Reference variables are defined like regular variables, except there is a(n) & in front of the name.
21. Reference variables allow arguments to be passed by ____________.
22. The exit function causes a program to terminate immediately.
23. Two or more functions may have the same name, as long as their __________ are different.

24. What is the advantage of breaking your application’s code into several small functions?

25. What is the difference between an argument and a parameter variable?

26. When a function accepts multiple arguments, does it matter what order the arguments are passed in?

27. What does it mean to overload a function?

28. If you are writing a function that accepts an argument and you want to make sure the function cannot change the value of the argument, what should you do?

29. Give an example where an argument should be passed by reference.

30. How do you return a value from a function?

31. Can a function have a local variable with the same name as a global variable?

32. When should a static local variable be used?

Algorithm Workbench

33. The following statement calls a function named half, which returns a value that is half that of the argument passed to it. Assume that result and number have both been defined to be double variables. Write the half function.

```
                    result = half(number);
```

```
double half (double n)
  {
      n / 2
      return n;
      }
```

34. A program contains the following function.

```
          int cube(int num)
          {
                    return num * num * num;
          }
```

Write a statement that passes the value 4 to this function and assigns its return value to the variable result.

```
result = cube(4);
```

35. Write a function, named timesTen, that accepts an integer argument. When the function is called, it should display the product of its argument multiplied times 10.

```
void timesTen(int i)
    {
       cout << (i * 10);
```

36. A program contains the following function.

```
void display(int arg1, double arg2, char arg3)
{
        cout << "Here are the values: "
        << arg1 << " " << arg2 << " " << arg3 << endl;
}
```

Write a statement that calls the function and passes the following variables to it:

```
int age;
double income;
char initial;
```

```
display ( age , income , initial);
```

37. Write a function named getNumber, which uses a reference parameter to accept an integer argument. The function should prompt the user to enter a number in the range of 1 through 100. The input should be validated and stored in the parameter variable.

```
void getNumber (int &num)
{
    cout << "please ent num";
      cin >> num;
       if ( not 1-100)
```

38. Write a function named biggest that receives three integer arguments and returns the largest of the three values.

Find the Errors

39. Each of the following functions has errors. Locate as many errors as you can.

A)
```
void total(int value1, value2, value3)
    {
        return value1 + value2 + value3;
    }
```

B)
```
double average(int value1, int value2, int value3)
    {
        double average;
        average = value1 + value2 + value3 / 3;
    }
```

C)
```
void area(int length = 30, int width)
    {
    return length * width;
    }
```

D)
```
void getValue(int value&)
    {
    cout << "Enter a value: ";
    cin >> value&;
    }
```

E)
```
// Overloaded functions
    int getValue()
    {
        int inputValue;
        cout << "Enter an integer: ";
        cin >> inputValue;
        return inputValue;
    }
    double getValue()
    {
    double inputValue;
        cout << "Enter a floating-point number: ";
        cin >> inputValue;
        return inputValue;
    }
```

CHAPTER 7 REVIEW / STUDY QUESTIONS

Fill-in-the-Blank and Short Answer

1. What does ADT stand for?

2. Which of the following must a programmer know about an ADT to use it?
A) What values it can hold
B) What operations it can perform
C) How the operations are implemented

3. The two common programming methods in practice today are _________ and _________.

4. ____________ programming is centered around functions, or procedures, whereas ____________ programming is centered around objects.

5. An object is a software entity that combines both __________ and ____________in a single unit.

6. An object is a(n) ______________ of a class.

7. Creating a class object is often called _____________ the class.

8. Once a class is declared, how many objects can be created from it?
A) 1
B) 2
C) Many

9. An object's data items are stored in its ____________________________________.

10. The procedures, or functions, an object performs are called its _________________.

11. Bundling together an object's data and procedures is called ___________________.

12. An object's members can be declared public or private.
A public member can be accessed by ____________________________________.
A private member can be accessed by ____________________________________.

13. Normally a class's ________________________________ are declared to be private and its ____ are declared to be public.

14. A class member function that uses, but does not change, the value of a member variable is called a(n) ___________.

15. A class member function that changes the value of a member variable is called a(n) ____________.

16. When a member function's body is written inside a class declaration, the function is a(n) _________ function.

17. A class constructor is a member function with the same name as the _________.

18. A constructor is automatically called when an object is _________.

19. Constructors cannot have a(n) _________ type.

20. A(n) _________ constructor is one that requires no arguments.

21. A destructor is a member function that is automatically called when an object is _________.

22. A destructor has the same name as the class, but is preceded by a(n) _________ character.

23. A constructor whose parameters all have default values is a(n) _________ constructor.

24. A class may have more than one constructor, as long as each has a different _________.

25. A class may only have one default _________ and one _________.

26. In general it is considered good practice to have member functions avoid doing _________.

27. When a member function forms part of the interface through which a client program can use the class, the function must be _____________.

28. When a member function performs a task internal to the class and should not be called by a client program, the function should be made ___________.

29. True or false: A class object can be passed to a function, but cannot be returned by a function.

30. True or false: C++ class objects are always passed to functions by reference.

31. It is considered good programming practice to store the declaration for a class, its function definitions, and the client program that uses the class in __________________ files.

32. If you were writing a class declaration for a class named Canine and wanted to place it in its own file, what should you name the file? ___________________

33. If you were writing the definitions for the Canine class member functions and wanted to place these in their own file, what should you name the file? ____________________

34. A structure is like a class, but normally only contains member variables and no _______________________.

35. By default, are the members of a structure public or private? _____________________

36. Before a structure variable can be created, the structure must be __________________.

37. When a structure variable is created its members can be initialized with either a(n) ________________________ or a(n) __________________________.

38. The __________ operator is used to access structure members.

39. An Inventory structure is declared as follows:

```
struct Inventory
{
        int itemCode;
        int qtyOnHand;
};
```

Write a definition statement that creates an Inventory variable named trivet and initializes it with an initialization list so that its code is 555 and its quantity is 110.

40. A Car structure is declared as follows:

```
struct Car
{
        string make,
        model;
        int year;
        double cost;
        Car(string mk, string md, int y, double c)
        { make = mk; model = md; year = y; cost = c; }
};
```

Write a definition statement that defines a Car structure variable initialized with the following information:

Make: Ford   Model: Mustang
Year: 2010   Cost: $22,495

41. Declare a structure named TempScale, with the following members:

fahrenheit: a double
celsius: a double

Next, declare a structure named Reading, with the following members:

windSpeed: an int
humidity: a double
temperature: a TempScale structure variable

Next, define a Reading structure variable named today.

Now write statements that will store the following data in the Reading variable.

Wind speed: 37 mph
Humidity: 32%
Fahrenheit temperature: 32 degrees
Celsius temperature: 0 degrees

42. Write a function called showReading. It should have a parameter that accepts a Reading structure variable (see question 41) and should display the values of the structure's member variables on the screen.

43. Write a function called inputReading that has a parameter to accept a Reading structure reference variable (see question 41). The function should ask the user to enter values for each member of the structure.

44. Write a function called getReading, which returns a Reading structure (see question 41). The function should ask the user to enter values for each member of a Reading structure, and then return the structure.

45. Write the declaration of a union called Items with the following members. Then define an Items union variable named anItem.

alpha: a character // 1 byte
num: an integer // 4 bytes
bigNum: a long integer // 4 bytes
real: a double // 8 bytes

46. How many bytes of memory will be allocated for anItem?

Algorithm Workbench

47. Assume a class named Inventory keeps track of products in stock for a company. It has member variables prodID, prodDescription, and qtyInStock. Write a constructor that initializes a new Inventory object with the values passed as arguments, but which also includes a reasonable default value for each parameter.

48. Write a remove member function that accepts an argument for a number of units and removes that number of units of an item from inventory. If the operation is completed successfully it should return the number of units remaining in stock for that item. However, if the number of units passed to the function is less than the number of units in stock, it should not make the removal and should return –1 as an error signal.

Find the Errors

Each of the following declarations, programs, and program segments has errors. Locate as many as you can.

49.

A)

```
struct
{ int x;
double y;
};
```

B)

```
struct Values
{ string name;
int age;
}
```

50.

A)

```
struct TwoVals
{
int a, b;
};
int main()
{
        TwoVals.a = 10;
        TwoVals.b = 20;
        return 0;
}
```

B)

```
#include <iostream>
using namespace std;
struct ThreeVals
{
        int a, b, c;
        void ThreeVals()
        {a = 1; b = 2; c = 3;}
};
int main()
{
        ThreeVals vals;
        cout << vals << endl;
        return 0;
}
```

51.

A)

```
struct Names
{       string first;
        string last;
};
int main()
{
        Names customer ("Smith", "Orley");
        cout << Names.first << endl;
        cout << Names.last << endl;
        return 0;
}
```

B)

```
struct TwoVals
{
        int a = 5;
        int b = 10;
};
int main()
{
        TwoVals v;
        cout << v.a << " " << v.b;
        return 0;
}
```

52.

A)

```
class Circle:
{
        private
        double centerX;
        double centerY;
        double radius;
        public
        setCenter(double, double);
        setRadius(double);
}
```

B)

```
#include <iostream>
using namespace std;
Class Moon;
{
Private;
        double earthWeight;
        double moonWeight;
Public;
        moonWeight(double ew);// Constructor
        { earthWeight = ew; moonWeight = earthWeight / 6; }
        double getMoonWeight();
        { return moonWeight; }
}
int main()
{
        double earth;
        cout >> "What is your weight? ";
        cin << earth;
        Moon lunar(earth);
        cout << "On the moon you would weigh "
        <<lunar.getMoonWeight() << endl;
        return 0;
}
```

53.

A)

```
#include <iostream>
using namespace std;
class DumbBell;
{
int weight;
public:
void setWeight(int);
};
void setWeight(int w)
{ weight = w; }
int main()
{
DumBell bar;
DumbBell.setWeight(200);
cout << "The weight is " << bar.weight << endl;
return 0;
}
```

B)

```
class Change
{
private:
int pennies;
int nickels;
int dimes;
int quarters;
Change()
        { pennies = nickels = dimes = quarters = 0; }
Change(int p = 100, int n = 50, d = 50, q = 25);
};

void Change::Change(int p, int n, d, q)
{
        pennies = p;
        nickels = n;
        dimes = d;
        quarters = q;
}
```

CHAPTER 8 REVIEW / STUDY QUESTIONS

Review Questions and Exercises

Fill-in-the-Blank and Short Answer

1. The _________ indicates the number of elements, or values, an array can hold.
2. The size declarator must be a(n) _________ with a value greater than _________.
3. Each element of an array is accessed and indexed by a number known as a(n) _________.
4. Subscript numbering in C++ always starts at _________.
5. The number inside the brackets of an array definition is the _________, but the number inside an array's brackets in an assignment statement, or any other statement that works with the contents of the array, is the _________.
6. C++ has no array _________ checking, which means you can inadvertently store data past the end of an array.
7. Starting values for the elements of an array may be specified with a(n) _________ list.
8. If a numeric array is partially initialized, the uninitialized elements will bet set to _________.
9. If the size declarator of an array definition is omitted, C++ counts the number of items in the _________ to determine how large the array should be.
10. To allow an array of structures or an array of objects to be initialized, the struct or class declaration should include a(n) _________.
11. By using the same _________ for multiple arrays, you can build relationships between the data stored in the arrays. These arrays are referred to as parallel arrays.
12. You cannot use the _________ operator to copy data from one array to another in a single statement.
13. Arrays are never passed to functions by _________ because there would be too much overhead in copying all the elements.
14. To pass an array to a function, pass the _________ of the array.
15. A(n) _________ array is like several arrays of the same type put together.
16. It's best to think of a two-dimensional array as having _________ and _________.
17. To define a two-dimensional array, _________ size declarators are required.
18. When initializing a two-dimensional array, it helps to enclose each row's initialization list in _________.
19. When a two-dimensional array is passed to a function, the number of _________ must be specified.
20. To print out all elements of a two-dimensional array you would normally use a(n) _________ loop.

21. Look at the following array definition.

```
int values[10];
```

A) How many elements does the array have?
B) What is the subscript of the first element in the array?
C) What is the subscript of the last element in the array?
D) If an int uses four bytes of memory, how much memory does the array use?

22. Given the following array definition:

```
int values[5] = { 4, 7, 6, 8, 2 };
```

What does the following statement display?

```
cout << values[4] << " " << (values[2] + values[3])
<< " " << ++values[1] << endl;
```

23. Look at the following array definition.

```
int numbers[5] = { 1, 2, 3 };
```

A) What value is stored in numbers[2]?
B) What value is stored in numbers[4]?

24. Assume that array1 and array2 are both 25-element integer arrays. Indicate whether each of the following statements is legal or illegal.

A) array1 = array2;
B) cout << array1;
C) cin >> array2;

25. When you pass an array name as an argument to a function, what is actually being passed?

26. How do you establish a parallel relationship between two or more arrays?

27. Look at the following array definition.

```
        double sales[8][10];
```

A) How many rows does the array have?
B) How many columns does the array have?
C) How many elements does the array have?
D) Write a statement that stores 3.52 in the last column of the last row in the array.

Use the following Car structure declaration to answer questions 28–30.

```
        struct Car
        {
                string make,
                model;
                int year;
                double cost;

                // Constructors
                Car()
                { make = model = ""; year = cost = 0; }

                Car(string mk, string md, int yr, double c)
                { make = mk; model = md; year = yr; cost = c; }
        };
```

28. Define an array named collection that holds 25 Car structures.

29. Define an array named forSale that holds 35 Car structures. Initialize the first three elements with the following data:

Make	Model	Year	Cost
Ford	Taurus	2006	$21,000
Honda	Accord	2004	$11,000
Jeep	Wrangler	2007	$24,000

30. Write a loop that will step through the array you defined in question 29, displaying the contents of each element.

Algorithm Workbench

31. The arrays array1 and array2 each hold 25 integer elements. Write code that copies the values in array1 to array2.

32. The following code totals the values in each of two arrays described in question 31. Will the code print the correct total for both arrays? Why or why not?

```
int total = 0; // Accumulator
int count; // Loop counter

// Calculate and display the total of the first array.
for (count = 0; count <= 25; count++)
        total += array1[count];
cout << "The total for array1 is " << total << endl;

// Calculate and display the total of the second array.
for (count = 0; count <= 25; count++)
        total += array2[count];
cout << "The total for array2 is " << total << endl;
```

33. In a program you need to store the identification numbers of 10 employees (as ints) and their weekly gross pay (as doubles).

A) Define two arrays that may be used in parallel to store the 10 employee identification numbers and 10 weekly gross pay amounts.

B) Write a loop that uses these arrays to print each employee’s identification number and weekly gross pay.

34. Revise your answer for question 33 to define and use an array of Payroll structures, instead of two parallel arrays. A Payroll structure should hold an employee ID and weekly gross pay amount.

35. In a program you need to store the names and populations of 12 countries. Create an appropriate array to store this information and then write the code needed to read the information into the array from a file named pop.dat.

36. A weather analysis program uses the following array to store the temperature for each hour of the day on each day of a week.

```
int temp[7][24];
```

Each row represents a day (0 = Sunday, 1 = Monday, etc.) and each column represents a time (0 = midnight, 1 = 1 a.m., … , 12 = noon, 13 = 1 p.m., etc.).

A) Write code to find Tuesday’s average temperature.
B) Write code to find the average weekly noon temperature.

Find the Errors

37. Each of the following definitions has errors. Locate as many as you can.

```
A)      int size;
        double values[size];
B)      int collection[-20];
C)      int hours[3] = 8, 12, 16;
```

38. Each of the following definitions has errors. Locate as many as you can.

```
A) int numbers[8] = {1, 2, , 4, , 5};
B) double ratings[];
C) values[3] = {6, 8.2, 'A'};
```

39. Each of the following functions contains errors. Locate as many as you can.

A)

```
void showValues(int nums)
{
    for (int count = 0; count < 8; count++)
        cout << nums[count];
}
```

B)

```
void showValues(int nums[4][])
{
    for (rows = 0; rows < 4; rows++)
        for (cols = 0; cols < 5; cols++)
            cout << nums[rows][cols];
}
```

CHAPTER 10 REVIEW / STUDY QUESTIONS

Fill-in-the-Blank and Short Answer

1. Each byte in memory is assigned a unique __________.
2. The __________ operator can be used to determine a variable’s address.
3. __________ variables are designed to hold addresses.
4. The __________ operator can be used to work with the variable a pointer points to.
5. Array names can be used as __________ and vice versa.
6. Creating variables while a program is running is called __________.
7. The __________ operator is used to dynamically allocate memory.
8. If the new operator cannot allocate the amount of memory requested, it throws __________.
9. A pointer that contains the address 0 is called a(n) __________ pointer.
10. When a program is finished with a chunk of dynamically allocated memory, it should free it with the __________ operator.
11. You should only use the delete operator to deallocate memory that was dynamically acquired with the __________ operator.
12. What does the indirection operator do?

13. Look at the following code.

```
int x = 7;
int *ptr = &x;
```

What will be displayed if you send the expression *iptr to cout? What happens if you send the expression ptr to cout?

14. Name two different uses for the C++ operator *.

15. Which arithmetic operations can be applied to pointers?

16. Assuming that ptr is a pointer to an int, what happens when you add 4 to it?

17. Look at the following array definition.

```
int numbers [] = {2, 4, 6, 8, 10};
```

What will the following statement display?

```
cout << *(numbers + 3) << endl;
```

18. What is the purpose of the new operator?

19. What happens when a program uses the new operator to allocate a block of memory, but the amount of requested memory isn’t available? How do programs written with older compilers handle this?

20. Under what circumstances can you successfully return a pointer from a function?

21. What is the purpose of the delete operator?

22. What is the difference between a pointer to a constant and a constant pointer?

23. Show C++ code for defining a variable ptr that is a pointer to a constant int.

24. Show C++ code for defining a variable ptr that is a constant pointer to int.

C++ Language Elements

25. Consider the function

```
void change(int *p)
{
    *p = 20;
}
```

Show how to call the change function so that it sets the integer variable

```
int i;
```

to 20.

26. Consider the function

```
void modify(int & x)
    {
        x = 10;
    }
```

Show how to call the modify function so that it sets the integer

```
int i;
```

to 10.

Algorithm Workbench

27. Write a function whose prototype is

```
void exchange(int *p, int *q);
```

that takes two pointers to integer variables and exchanges the values in those variables.

28. Write a function

```
void switchEnds(int *array, int size);
```

that is passed the address of the beginning of an array and the size of the array. The function swaps the values in the first and last entries of the array.

Predict the Output

29. Given the variable initializations

```
int a[5] = {0, 10, 20, 30, 40};
int k = 3;
int *p = a + 1;
```

determine the output from each of the following statements:

A) cout << a[k];

B) cout << *(a + k);

C) cout << *a;

D) cout << a[*a];

E) cout << a[*a + 2];

F) cout << *p;

G) cout << p[0];

H) cout << p[1];

Find the Error

30. Each of the following declarations and program segments has errors. Locate as many as you can.

```
A)      int ptr*;

B)      int x, *ptr;
        &x = ptr;

C)      int x, *ptr;
        *ptr = &x;

D)      int x, *ptr;
        ptr = &x;
        ptr = 100; // Store 100 in x
        cout << x << endl;

E)      int numbers[] = {10, 20, 30, 40, 50};
        cout << "The third element in the array is ";
        cout << *numbers + 3 << endl;

F)      int values[20], *iptr;
        iptr = values;
        iptr *= 2;

G)      double level;
        int dPtr = &level;

H)      int *iptr = &ivalue;
        int ivalue;

I)      void doubleVal(int val)
        {
                *val *= 2;
        }

J)      int *pint;
        new pint;

K)      int *pint;
        pint = new int;
        pint = 100;
```

L)

```
int *pint;
pint = new int[100]; // Allocate memory
.
Process the array
.
delete pint;// Free memory
```

M)

```
int *getNum()
{
        int wholeNum;
        cout << "Enter a number: ";
        cin >> wholeNum;
        return &wholeNum;
}
```