

Unless otherwise stated within a problem, assume that each program and/or program segment compiles and runs without error. Each question is independent unless otherwise stated. No calculators are allowed.

1. True or False The following code segment generates an error message.

```
int A[10];  
int B[5];  
B[-5] = 1;
```
2. True or False A new class that is created from an existing class using inheritance is called a derived class. The parent class is called the base class.
3. True or False When an object that is an instance of a derived class is instantiated, the constructor for the base class is invoked before the body of the constructor for the derived class is invoked.
4. True or False Initializing data members of a class using a data member initialization list is more efficient than calling a mutator from the constructor.
5. True or False The relationship has-a indicates inheritance. For example, a car has-a steering wheel.
6. True or False The is-a relationship is transitive. A Siamese cat is a type of cat, a cat is a mammal, therefore a Siamese cat is a mammal.
7. True or False An array must be a value parameter.
8. True or False An object that contains an array as a data member may be a value parameter.
9. True or False An array element may be either a value parameter or reference parameter.
10. True or False Arrays are first-class objects.
11. True or False If a mixed-mode arithmetic expression involves integral and floating-point operands, the floating-point operand is truncated and the appropriate integer operation is performed.
12. True or False A `continue` statement indicates that the body of the innermost loop that contains the statement is finished for the current iteration.
13. True or False For the arithmetic operators, the precedence from highest to lowest is unary plus and minus; then multiplication, division, and remainder; and then addition and subtraction.
14. True or False The versions of the copy constructor and member assignment operator supplied by the compiler perform deep copying.
15. True or False The size of an array is given as a bracketed expression whose value is derived from run-time constants.
16. True or False In an array, each element has its own subscript value. The first element in the array has a subscript of 1, the second element has a subscript of 2, and so on.
17. True or False An array element is a first-class object.
18. True or False The elements of an array can be stored in noncontiguous memory.
19. True or False In `C++` multidimensional arrays, the array elements are always stored in row-major order.
20. True or False A binary digit is called a byte.
21. True or False A bit is eight bytes.

22. True or False Data members represent the properties or attributes of a class.
23. True or False The statement `#include <iostream.h>` is a preprocessor directive that in part defines the error stream `cerr`.
24. True or False The C++ object type `char` is used to represent a character.
25. True or False Characters are always encoded using the ASCII character set.
26. True or False A C++ name consists of a sequence of letters, digits, and underscores. A valid name can only begin with a letter.
27. True or False C++ names are not case sensitive.
28. True or False Integer division can produce a floating-point result.
29. True or False The assignment operator is right associative and has lower precedence than the arithmetic operators.
30. True or False Elements of a local array whose base type is a numeric type are initialized by default to 0.
31. What is printed by this program statement?
`cout << 20 /* 10 */ / 5;`
32. What is the decimal value of the largest integer that can be stored in a 16-bit unsigned `int` variable? No calculators are allowed.
33. What values are assigned to the variables `i`, `j`, and `k` after execution of this program segment?
`int i = 10;
int j = 20;
int k = 30;
i = (j = (k = k + 7) - 3) + 4;`
34. Show the 16-bit, two's-complement representation of each of the following C++ integer constants. No calculators are allowed.
- (a) 59
 - (b) -39
 - (c) -32768
 - (d) 01276
 - (e) 0xABCD
35. The following object definitions are in effect for all parts of this question:
`bool P = true;
bool Q = false;
bool R = true;
bool S = false;`
- Evaluate each of the following expressions:
- (a) `R && !P || Q`
 - (b) `P != Q && S < R`
 - (c) `!(3 < 4) || 6 < 7 && 9 > 8`
 - (d) `2 + 2 == 2 * 2`
 - (e) `P == Q || R == S`

36. What value is printed by this program statement?

```
cout << - 2 - 3 * 4 / 5 + 6 % 7;
```

37. What value is printed by this program statement?

```
cout << 3 / 4 * 5 + 6 * 8 / 4;
```

38. What values are assigned to variables x, y, and z after execution of this program segment?

```
int i = 10;
int j = 20;
int k = 30;
int x, y, z;
x = i * j % k;
y = i + 1 * j - 1 / k;
z = k / 3 % 3 + k / i - 1;
```

39. The following object definitions are in effect for all parts of this question.

```
int i = 5;
int j = 6;
bool a = true;
bool b = false;
float r = 3.1;
float s = 2.7;
```

What is printed by each program segment below?

(a)

```
if (i <= j) if (b < a) if (r < s) cout << 1 << endl;
else cout << 2 << endl; else cout << 3 << endl;
else cout << 4 << endl;
```

(b)

```
if (r < s) cout << 1 << endl; else if (i != j) cout << 2 << endl;
else if (a == b) cout << 3 << endl; else cout << 4 << endl;
```

40. What is printed for the following program segment?

```
int count1 = 0;
int count2 = 0;
int count3 = 0;
for (int i = 1; i <= 10; i++) {
    count1 += 1;
    for (int j = 0; j < 10; j++) {
        count2 += 1;
        for (int k = -10; k < 0; k++) {
            ++count3;
        }
    }
}
cout << "count1 =" << count1 << endl;
cout << "count2 =" << count2 << endl;
cout << "count3 =" << count3 << endl;
```

41. Suppose the following definitions are in effect.

```
class A {
    public:
        A();
        void f();
        int value1;
    protected:
        int value2;
    private:
        int value3;
};
class B : public A {
    // ...
    protected:
        void g();
    // ...
};
```

Indicate which of the numbered lines (if any) of the following functions compile correctly:

```
void A::f() {
    A a;
L1.     a.value1 = 1;
L2.     a.value2 = 2;
L3.     a.value3 = 3;
}
void B::g() {
    B b;
L4.     b.value1 = 1;
L5.     b.value2 = 2;
L6.     b.value3 = 3;
}
void MyFunction() {
    B b;
L7.     b.value1 = 1;
L8.     b.value2 = 2;
L9.     b.value3 = 3;
}
```

42. What is printed by the following program fragment?

```
int j;
for (int i = 0; i < 100; ++i) {
    j = i;
}
cout << j << endl;
```

43. For the following program:

```
int main() {
    char c = 'c';
    char d = 'd';
    cin >> c;
    cin >> d;
    cout << "c = " << c << " d = " << d << endl;
    return 0;
}
```

(a) What is the output if the standard input stream contains the following?

```
12
A
B
```

(b) What is the output if the standard input stream contains the following?

```
w
x
y
z
```

44. What is printed by the following program segment?

```
int i = 10;
int j = 20;
{
    int i = 30;
    int j = 40;
}
cout << "i = " << i << " j = " << j << endl;
```

45. Member functions that return the value of an attribute of an object are called _____.

46. Member functions that set or change the value of an attribute of an object are called _____.

47. One of the standard libraries defines the _____ macro that enables an integral/boolean expression to be evaluated. If the expression is false, the program is terminated.

48. A logical expression that is being evaluated is subject to the _____ rule that states that once the overall value of an expression is known, evaluation ceases.

49. When defining a function with an array parameter, the _____ parameter definition does not need to include the size of the first dimension.

50. What is printed by each program statement?

- (a) `cout << ceil(3.14159);`
- (b) `cout << fabs(-2);`
- (c) `cout << floor(3.14159);`
- (d) `cout << pow(3, 3);`

51. Suppose we want to sort 4 numbers. How many different orderings can there be?

52. Consider a program that in part reads in 75 character values into the array `message` using the statements given below:

```
int i;
const int n = 75;
char message[n];
for (i = 0 ; i < n; ++i) {
    get(message[i]);
}
```

Complete the following function definition.

```
void get(_____) {
    cin >> b;
    return;
}
```

53. Give the truth table for the following logical expression: $(\text{not } P) \text{ or } Q$

54. What operator is used to refer to a particular element of an array?

55. Suppose that you run the following C++ program:

```
#include <iostream.h>
// prototype of function f() here
int main()
    int first = 1;
    int second = 2;
    second = f(first);
    cout << first << " and " << second << endl;
    return 0;
}
```

and obtain the following output:

```
2 and 76
```

What should the prototype of function `f()` be?

56. What is the output of the following code segment?

```
void f(int a[]) {
    int i;
    for (i = 0; a[i] > 0; ++i) {
        cout << a[i] << " ";
    }
    cout << endl;
    return;
}
int main() {
    int a[9] = {10, 9, 8, 7, 6, -3, 3, 1, -1};
    int b[7] = {6, 4, 2, 0, -2, 7, 5};
    f(a);
    f(b);
    return 0;
}
```

57. Describe the output of the following code segment.

```
char c = 65;
cout << ++c;
```

58. Why do we tend to use named constants rather than literals when defining the size of an array or a vector?

59. What is meant by row-major order?

60. Write the single C++ statement that defines `Suit` to be an enumeration type whose possible values are heart, diamond, club, and spade.

61. Suppose the following global definitions are in effect.

```
class A {
public:
    A(int v1 = 0, int v2 = 0);
    int get1() const;
    int get2() const;
    int set1(int v1);
    int set2(int v2);
protected:
    void f1();
private:
    int value1;
    int value2;
    void f2();
}
A a(1, 1);
const A b(1, 2);
```

- Does `A` have a copy constructor? Why?
- Does `A` have a default constructor? Why?
- Can function `main()` invoke `a.set1(3)`? Why?
- Can function `main()` invoke `b.set1(3)`? Why?
- Can function `main()` invoke `a.f1()`? Why?
- Can `A::f1()` invoke `A::f2()`? Why?
- A non-`A` function `g()` invokes `A::f1()`. Must `g()` have some particular characteristic? If so, what is it?

62. Complete the following table describing inheritance access rights.

Inheritance type	Base class member access	Derived class member access
public	public	
	protected	
	private	
protected	public	
	protected	
	private	
private	public	
	protected	
	private	

63. The array `X` contains n floating point numbers. The function `Sort()`, listed below, was intended to sort the array `X` into non-decreasing order when invoked like this:

```
float X[] = {3.1, 9.9, -4.5, 0.0, 7.8, 8.4, 5.5, 8.1, 9.0, -0.3};
Sort (X, 10);
```

Although the function `Sort()` compiles without error, it does not produce the desired result of rearranging the array into non-decreasing order. Examine the code below and identify the line causing the error, and then rewrite that one line of code such that the error is corrected.

```
L1. void Sort (float a[], int n) {
L2.     for (int j = 0; j < n-1; ++j) {
L3.         for (int k = j+1; k <= n; ++k) {
L4.             if (a[j] > a[k]) {
L5.                 float temp = a[j];
L6.                 a[j] = a[k];
L7.                 a[k] = temp;
L8.             }
L9.         }
L10.     }
L11. }
```

64. The factorial of a positive integer n is the number $n!$ obtained by multiplying n by all of the positive integers less than n :

$$n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$$

Write the function `Fact()` that computes the factorial of its integer parameter.

65. A leap year is a year in which an extra day (February 29) is added to the regular calendar. The rule for determining whether a year is a leap year is:

A year is a leap year if it evenly divisible by 4, except that centennial years (those evenly divisible by 100) are not leap years, except that centennial years evenly divisible by 400 are leap years. For example, 1800 and 1900 are not leap years, 1992 and 1996 are leap years, and 2000 is a leap year.

Write a Boolean function `Leap()` that returns true if its integer parameter y is a leap year and returns false if it is not a leap year.

66. The two-dimensional matrix `Temp` contains r rows and c columns of double-precision floating-point numbers, each number representing a temperature in degrees Fahrenheit (r and c are global constants). Write a function `Average()` that computes the average of all the temperatures in matrix `Temp`. Your function must be compatible with this invocation:

```
cout << "Average temperature is: " << Average(Temp, r, c) << " degrees.";
```

67. The ancient Babylonians discovered that the square root of 2 could be approximated by making an estimate of the square root of 2 (call it x), and then repeatedly iterating this expression:

$$x = (x + (2/x)) / 2$$

in which the old estimate of x is used on the right-hand side of the assignment statement to produce a new value of x , then the new value of x is taken as the old value of x and the calculation is repeated. If the calculation is repeated a sufficient number of times, the value of x converges to the value of the square root of 2.

Write a double-precision floating-point function `Root2()` which uses the Babylonian algorithm to return an approximation to the square root of two. Use an initial approximation of $x=2$. Stop the iteration when the absolute difference between x^2 and 2.0 is less than epsilon, where epsilon equals $5E-8$.

68. Write a flexible code segment that accomplishes the following:

- (a) Defines a constant `MaxSize` equal to 10.
 - (b) Defines an array `List` whose base type is integer that can represent at most `MaxSize` values.
 - (c) Sets the second element of `List` to the value 12.
 - (d) Sets the last element of `List` to the value 66.
 - (e) Is the value 3.1415 a legal element value for `List` ?
 - (f) Is the value 3.1415 a legal subscript for `List` ?
69. Develop a class `Date` for representing a calendar. The class should provide a default constructor which initializes the date to September 14, 1752, and has three optional integer parameters corresponding to the desired month, day, and year. The class should have three public inspectors and three public mutators that allow the month, day, and year to be accessed. The operator `++` should be auxiliary overloaded so that when applied to a `Date` object, the object's new value is the successive day. Also prototype auxiliary functions `Tostring()` that returns a string version of its `Date` parameter, and `DayOfWeek()` that returns the day of the week on which its `Date` parameter falls. The return type for `DayOfWeek()` should be an enumerated type `Days` whose symbolic constants are `Sunday`, `Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, and `Saturday`.
70. Define a `bool` function `Equal()` with three parameters: `A`, `B`, and `n`. Parameters `A` and `B` are `int` arrays; `int` parameter `n` is the size of the arrays. The function iteratively compares the elements of the arrays. If each of the `n` pairs of elements is the same, the function returns `true`; otherwise, the function returns `false`.

The following is the class definition for `Rational`.

```
class Rational {
public:
    Rational();
    Rational(int numer, int denom = 1);
    Rational Add(const Rational &r) const;
    Rational Multiply(const Rational &r) const;
    Rational Equals(const Rational &r) const;
    void Insert(ostream &sout) const;
    void Extract(istream &sin);
protected:
    int Numerator() const;
    int Denominator() const;
    void SetNumerator(int numer);
    void SetDenominator(int denom);
private:
    int NumeratorValue;
    int DenominatorValue;
};
```

71. Define a `Rational` member function `FloatingPoint()` that returns a single-precision floating-point representation of the invoking object. Your function should practice information hiding.
72. Define a `Rational` facilitator member function `Equals` (see prototype above) that returns `true` if two rational numbers are numerically equivalent, and `false` otherwise. Thus, `Equals` returns `true` when comparing `1/2` and `2/4`.