



**READ THIS NOW!**

Failure to read and follow the instructions below may result in severe penalties.

- Print your name in the space provided below.
- Print your name and ID number on the Opscan form and code your ID number and **Form B** correctly on the Opscan form.
- Choose the single best answer for each question — some answers may be partially correct. If you mark more than one answer to a question, you will receive no credit for any of them.
- Unless a question involves determining whether given C++ code is syntactically correct, it should be; if you think there is a syntax error where there should not be, ask. Unless a question specifically deals with compiler #include directives, you should assume the necessary header files have been included.
- Be careful to distinguish integer values from floating point values (containing a decimal point). In questions/answers that require a distinction between integer and real values, integers will be represented without a decimal point, whereas real values will have a decimal point, [1044 (integer), 1044.0 (floating point)].
- **This is a closed-book, closed-notes examination.**
- **No laptops, calculators or other electronic devices may be used during this examination.**
- **You may not discuss (in any form: written, verbal or electronic) the content of this examination with any student who has not taken it.**
- **You must return this test form when you complete the examination. Failure to adhere to any of these restrictions is an Honor Code violation.**
- There are 30 equal-valued multiple-choice questions.
- The answers you mark on the Opscan form will be considered your official answers.
- When you have finished, sign the pledge at the bottom of this page and turn in the test and your Opscan.

**Do not start the test until instructed to do so!**

Name (Last, First) \_\_\_\_\_ printed

**Pledge:** On my honor, I have neither given nor received unauthorized aid on this examination.

\_\_\_\_\_  
*Signature*

For the next two questions, assume the declarations below:

```

_____ MAXSIZE = 20;
char Buffer[3 * MAXSIZE];

```

1. How should the blank preceding MAXSIZE be filled?
 

1) char	4) const int	7) 2 or 4 only
2) int	5) All of them	8) None of these
3) const char	6) 1 or 3 only	
  
2. What is the range of valid index values for the array Buffer[ ]?
 

1) 1 through 60	4) 0 through 60	7) None of these
2) 1 through 61	5) 0 through 61	
3) 1 through 62	6) 0 through 62	

3. Assume the following declarations:

```

void Fix(double x, int& k);
int Count = 12;
double Area = 6.28;

```

Which of the following would represent logically and syntactically appropriate call of the function Fix?

- |                            |                                 |
|----------------------------|---------------------------------|
| 1) Fix(6.85, 24);          | 6) All of them are appropriate. |
| 2) Fix(6.85, Count);       | 7) 1 and 2 only                 |
| 3) Fix(Area, 24);          | 8) 2 and 5 only                 |
| 4) Fix(Area, Count + 5);   | 9) 2, 4 and 5 only              |
| 5) Fix(Area + 2.0, Count); | 10) None of these               |

For the next two questions, consider executing the following code fragment:

```

int j = 4;
while (j <= 32) {
    j = 2 * j;
    cout << j << endl;
}

```

4. How many times will the body of the loop be executed?
 

1) 0	3) 2	5) 4	7) None of these
2) 1	4) 3	6) infinite loop	
  
5. What is the last value printed?
 

1) Nothing is printed	3) 8	5) 32	7) infinite loop
2) 4	4) 16	6) 64	8) None of these

6. When a function is called, and a `string` variable is passed as a parameter to that function, how will the `string` variable be passed (by default)?

- 1) by reference
- 2) by value
- 3) by constant reference
- 4) `string` variables can't be parameters
- 5) None of these

7. In C++, which of the following are true about array variables?

- |   |                  |
|---|------------------|
| 1) The elements of an array must be of the same type. | 5) 1 and 2 only  |
| 2) The elements are accessed by location.             | 6) 1 and 3 only  |
| 3) Aggregate assignment is supported.                 | 7) 2 and 3 only  |
| 4) All of them  | 8) None of these |

8. A function, `G`, has two formal parameters, `P1` of type `int` and `P2` of type `char`. The data flow (communication) for variable `P1` is one-way, out of the function. The data flow for variable `P2` is one-way, into the function. Which of the following is the **most appropriate** prototype for `G`?

- |  |                  |
|--|------------------|
| 1) <code>void G(int P1, char P2);</code>           | 6) 1 and 2 only  |
| 2) <code>void G(int P1, char&amp; P2);</code>      | 7) 3 and 4 only  |
| 3) <code>void G(int&amp; P1, char P2);</code>      | 8) 1 and 3 only  |
| 4) <code>void G(int&amp; P1, char&amp; P2);</code> | 9) None of these |
| 5) All of them are equally appropriate.            |                  |

```
void Ring(int& Frodo, int Sam);
int Sauron = 3;

int main() {
    int Merry = 7, Pippin = 5;
    Ring(Merry, Pippin);
    cout << "Merry = " << Merry << endl;
    cout << "Pippin = " << Pippin << endl;
    cout << "Sauron = " << Sauron << endl;
    return 0;
}
```

```
void Ring(int& Frodo, int Sam) {
    Sauron = Frodo;
    Frodo = Frodo - 2 * Sam;
    Sam = Sauron;
}
```

9. What value is printed for the variable `Merry`?

- |       |      |       |                  |
|-------|------|-------|------------------|
| 1) -5 | 3) 3 | 5) 7  | 7) 21            |
| 2) -3 | 4) 5 | 6) 12 | 8) None of these |

10. What value is printed for the variable `Pippin`?

- |       |      |       |                  |
|-------|------|-------|------------------|
| 1) -5 | 3) 3 | 5) 7  | 7) 21            |
| 2) -3 | 4) 5 | 6) 12 | 8) None of these |

11. What value is printed for the variable `Sauron`?

- |       |      |       |                  |
|-------|------|-------|------------------|
| 1) -5 | 3) 3 | 5) 7  | 7) 21            |
| 2) -3 | 4) 5 | 6) 12 | 8) None of these |

Consider a data file that contains lines of score information, so that each line contains one or more integer values, followed by a zero that marks the end of the data on that line. The first line of the file will specify the number of data lines that follow.

For the next four questions, consider the function given below, which is intended to read a data file as described above and determine the largest sum that can be obtained from any data line.

```
int hiScore(ifstream& Data) {
    int maxSoFar = INT_MIN;           // 1
    int numLines, lineSum, Value;     // 2
    Data >> numLines;                 // 3
    for (int Line = 1; Line <= numLines; Line++) { // 4
        while ( Value != 0 ) {        // 5
            lineSum = lineSum + Value; // 6
        }                             // 7
    }                                  // 8
    return maxSoFar;                  // 9
}
```

**Data file:**

```
10
23 14 9 0
31 0
12 2 7 20 19 0
22 14 2 0
8 3 21 9 4 3 2 0
25 3 0
47 9 0
5 5 5 5 5 5 0
1 2 34 8 0
20 5 13 0
```

12. Where should the variable `lineSum` be set to 0?

- |           |            |                  |
|-----------|------------|------------------|
| 1) line 2 | 3) line 7  | 5) None of these |
| 2) line 5 | 4) line 10 |                  |

13. Where should the statement `Data >> Value` appear?

- |            |                       |                       |
|------------|-----------------------|-----------------------|
| 1) line 5  | 4) line 12            | 7) line 8 and line 10 |
| 2) line 8  | 5) line 5 and line 10 | 8) line 8 and line 12 |
| 3) line 10 | 6) line 5 and line 12 | 9) None of these      |

14. Where should the following `if`-statement appear: `if ( lineSum > maxSoFar )`  
`maxSoFar = lineSum;`

- |            |                              |                  |
|------------|------------------------------|------------------|
| 1) line 12 | 3) either line 12 or line 14 | 5) None of these |
| 2) line 14 | 4) It should NOT appear.     |                  |

15. What would happen if the input file contained more lines of data than were promised by its first line?

- 1) The function would process the extra lines of data.
- 2) The function would process as many lines as were promised, and return an answer based only on those lines.
- 3) The function would fail to terminate.
- 4) None of these

16. Consider the following function:

```
int H(int i, int j) {
    int Sum = 0;
    for ( ; i > 0; i = i - 1) {
        for (int k = 1; k <= j; k = 2 * k)
            Sum = Sum + i + k;
    }
    return Sum;
}
```

What value is returned from the call H(5, 3)?

- |      |       |       |                  |
|------|-------|-------|------------------|
| 1) 0 | 3) 16 | 5) 39 | 7) 45            |
| 2) 9 | 4) 23 | 6) 42 | 8) None of these |

For the next three questions, assume the following declarations:

```
const int SZ = 1000;
string Names[SZ];
```

17. The following loop is supposed to initialize all the cells of the array Names[] to hold "Blank".

```
for ( _____; _____; Pos++) {
    Names[Pos] = "Blank";
}
```

How should the two blanks in the loop header be filled?

- |   |                  |
|---|------------------|
| 1) <code>int Pos = 0</code> and <code>Pos &lt;= SZ</code> | 5) 1 or 3        |
| 2) <code>int Pos = 0</code> and <code>Pos &lt; SZ</code>  | 6) 2 or 4        |
| 3) <code>int Pos = 1</code> and <code>Pos &lt; SZ</code>  | 7) None of these |
| 4) <code>int Pos = 1</code> and <code>Pos &lt;= SZ</code> |                  |
18. The following loop is supposed to read data from an input stream, count the entries, and store it into the array Names[]. Assuming the input file is formatted to conform to the logic of the code, does the code contain a logic error?

```
int Read = 0;
In >> Names[Read];
while ( In && Read < SZ ) {
    Read++;
    In >> Names[Read];
}
```

- |   |                 |
|---|-----------------|
| 1) No, the code contains no logic errors.   | 5) 2, 3 and 4   |
| 2) Yes, if the input file contains more than 1000 entries, the code will try to store data past the end of the array. | 6) 2 and 3 only |
| 3) Yes, the code will not stop reading if the end of the file is reached.   | 7) 2 and 4 only |
| 4) Yes, if the input file contains fewer than 1000 entries, the code will change the first unused cell of the array.  | 8) 3 and 4 only |

19. The following loop is supposed to print all data stored in the array `Names []` by the code in the previous question (with any logic errors there corrected). What value should be used in the blank in the loop header?

```
for (int Pos = 0; Pos < _____; Pos++) {
    cout << Names[Pos] << endl;
}
```

- |           |             |                  |
|-----------|-------------|------------------|
| 1) SZ - 1 | 4) Read - 1 | 7) None of these |
| 2) SZ     | 5) Read     |                  |
| 3) SZ - 1 | 6) Read + 1 |                  |

20. Suppose the first few lines of a function are as follows:

```
string Punctuate(string FName, string LName) {
    string FullName = LName + COMMA + FName;
    . . .
}
```

Assuming the program compiles without any errors, the identifier `COMMA` must be:

- |   |                          |
|---|--------------------------|
| 1) local to the function <code>Punctuate</code>               | 5) Could be any of those |
| 2) a formal parameter in the function <code>Punctuate</code>  | 6) 1 or 2                |
| 3) an actual parameter to the function <code>Punctuate</code> | 7) 3 or 4                |
| 4) local to the function that calls <code>Punctuate</code>    | 8) None of these         |

For the next two questions, consider the following code fragment:

```
struct Name {
    string FName;
    string LName;
};
const int SIZE = 50;
. . .
int main() {
    Name List[SIZE];
    Name Staff;
    . . .
}
```

21. Which of the following are logically and syntactically valid function invocations for a function with the prototype:

```
void readName (Name& newPerson);
```

- |                                     |                                  |                    |
|-------------------------------------|----------------------------------|--------------------|
| 1) <code>readName(List);</code>     | 4) <code>readName(Staff);</code> | 7) 2 and 4 only    |
| 2) <code>readName(List[3]);</code>  | 5) All of them are valid.        | 8) 2, 3 and 4 only |
| 3) <code>readName(List[50]);</code> | 6) 2 and 3 only                  | 9) None of these   |

22. Which of the following are valid function invocations for a function with the prototype:

```
void writeStaff(const Name Employee[]);
```

- |                                      |                                     |                    |
|--------------------------------------|-------------------------------------|--------------------|
| 1) <code>writeStaff(List);</code>    | 4) <code>writeStaff(List[]);</code> | 7) 1, 2 and 3 only |
| 2) <code>writeStaff(List[3]);</code> | 5) All of them are valid.           | 8) 2, 3 and 4 only |
| 3) <code>writeStaff(Staff);</code>   | 6) 1 and 2 only                     | 9) None of these   |

For the next four questions, assume the following `struct` type declarations:

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

```
struct Rectangle {
    Point NW;
    int Width;
    int Height;
};
```

And, assume the following variable declarations:

```
Point Corner;
Rectangle R;
```

23. Which of the following code fragments would set `Corner` to represent the point (40, 100)?
- |  |                  |
|--|------------------|
| 1) <code>Corner.x = 40;</code><br><code>Corner.y = 100;</code> | 4) All of them   |
| 2) <code>x.Corner = 40;</code><br><code>y.Corner = 100;</code> | 5) 1 and 2 only  |
| 3) <code>x = 40;</code><br><code>y = 100;</code>               | 6) 1 and 3 only  |
|  | 7) 2 and 3 only  |
|  | 8) None of these |
24. Which of the following code fragments would calculate the area of the `Rectangle` variable `R` (assuming it's been initialized)?
- |  |                 |
|--|-----------------|
| 1) <code>Width * Height</code>                     | 4) All of them  |
| 2) <code>Rectangle.Height * Rectangle.Width</code> | 5) 1 and 2 only |
| 3) <code>R.Height * R.Width</code>                 | 6) 1 and 3 only |
|  | 7) 2 and 3 only |
|  | 8) None of them |
25. Assuming that `Corner` was set correctly in the earlier question, which of the following code fragments would set the `Point` member of `R` to be the point (40, 100)?
- |  |                     |
|--|---------------------|
| 1) <code>R.NW.x = 40;</code><br><code>R.NW.y = 100;</code>       | 4) All of them will |
| 2) <code>R.Point.x = 40;</code><br><code>R.Point.y = 100;</code> | 5) 1 and 2 only     |
| 3) <code>R.NW = Corner;</code>                                   | 6) 1 and 3 only     |
|  | 7) 2 and 3 only     |
|  | 8) None of these.   |
26. Which of the following would usefully print the value of the `Point` variable `Corner`?
- |   |
|---|
| 1) <code>cout &lt;&lt; '(' &lt;&lt; Corner.x &lt;&lt; ", " &lt;&lt; Corner.y &lt;&lt; ')';</code> |
| 2) <code>cout &lt;&lt; Corner;</code>   |
| 3) 1 and 2  |
| 4) None of them   |

For the next four questions, consider the incomplete function definition given below:

```
// Mapper traverses an array of integers from front to back, and compares each
// element to its successor, replacing the element with its successor if the
// successor is larger. For example, given the array:
//
//      9  17  23  21  4  7  10  9
//
// Mapper would transform it into the array:
//
//      17  23  23  21  7  10  10  9
//
// Parameters:
//      List[]  array to be reversed
//      Usage  number of values stored in List[]

void Mapper(int List[], int Usage) {           // Line 1
    _____;                               //      2
    while ( _____ ) {                     //      3
        if ( List[Pos] < _____ ) {      //      4
            _____;                       //      5
        }
        Pos++;                                 //      6
    }
}
```

27. How should the blank in Line 2 be filled to properly initialize the loop control logic?

- |                             |                                 |                             |
|-----------------------------|---------------------------------|-----------------------------|
| 1) <code>int Pos</code>     | 3) <code>int Pos = 1</code>     | 5) It should be left blank. |
| 2) <code>int Pos = 0</code> | 4) <code>int Pos = Usage</code> | 6) None of these            |

28. How should the blank in Line 3 be filled to properly terminate the loop?

- |                                    |  |                      |
|------------------------------------|--|----------------------|
| 1) <code>Pos &lt; Usage - 1</code> | 3) <code>Pos &lt;= Usage</code>              | 5) <code>true</code> |
| 2) <code>Pos &lt; Usage</code>     | 4) <code>List[Pos] &lt; List[Pos + 1]</code> | 6) None of these     |

29. How should the blank in Line 4 be filled to decide whether the list needs to be changed?

- |                               |                               |                      |
|-------------------------------|-------------------------------|----------------------|
| 1) <code>Pos</code>           | 3) <code>List[Pos + 1]</code> | 5) <code>Size</code> |
| 2) <code>List[Pos - 1]</code> | 4) <code>List[Usage]</code>   | 6) None of these     |

30. How should the blank in line 5 be filled to carry out the specified transformation of the list elements?

- |                                 |   |                       |
|---------------------------------|---|-----------------------|
| 1) <code>Pos = Pos + 1</code>   | 3) <code>List[Pos] = List[Pos + 1]</code> | 5) <code>break</code> |
| 2) <code>List[Pos] = Pos</code> | 4) <code>List[Pos + 1] = List[Pos]</code> | 6) None of these      |