



READ THIS NOW!

**Failure to read and follow the instructions below may result in severe penalties.
Failure to adhere to these directions will not constitute an excuse or defense.**

- Print your name and 4 digit class number in the space provided below.
- Print your name and ID number on the OpSCAN form; be sure to code your ID number on the OpSCAN form. Code **Form A** on the OpSCAN.
- Choose the single best answer for each question — some answers may be partially correct. If you mark more than one answer, it will be counted wrong.
- Unless a question involves determining whether given C++ code is syntactically correct, assume that it is. The given code has been compiled and tested, except where there are deliberate errors. Unless a question specifically deals with compiler #include directives, you should assume the necessary header files have been included.
- Note that questions about printed values disregard formatting completely.
- In questions/answers which 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 (real)].
- When you have completed the test, sign the pledge at the bottom of this page and turn in the test and your OpSCAN.
- **This is a closed-book, closed-notes examination. No 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 multiple-choice questions.
- Mark your answers on the test form and on the OpSCAN. The answers you mark on the OpSCAN form will be considered your official answers.

Do not start the test until instructed to do so!

Name _____

--	--	--	--

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

For the next two questions, choose which type of loop would be most appropriate for solving the following problems.
Choose from:

- | | | |
|----------------------------|---------------------------|-------------------------------|
| 1) a count-controlled loop | 2) a flag-controlled loop | 3) a sentinel-controlled loop |
| 4) an EOF-controlled loop | 5) either 2 or 3 | |

1) Problem: "Input an integer value, then print 'Happy Birthday' that many times."

2) Problem: "Count the number of positive integers in a data file of unknown length."

Consider the execution of the following fragment of code:

```
int length = 5;
int count = 4;
while (count <=6)
{
    if (length >= 100)
        length = length - 2;
    else
        length = count * length;
    count++;
}
```

3) After execution of the code above, what is the value of length?

- | | | | | |
|--------|--------|-------|-------|------------------|
| 1) 600 | 2) 100 | 3) 98 | 4) 20 | 5) none of these |
|--------|--------|-------|-------|------------------|
-

For the next three questions, assume the input file stream `ifile` is connected to a file containing the following data:

4 3 2 -4 5 7

Consider the execution of the code fragment given in each question and determine the value that would be printed.
Choose from the following answers:

- | | | | | |
|-------|-------|-------|------------------|-------|
| 1) 3 | 2) 4 | 3) 6 | 4) 9 | 5) 10 |
| 6) 13 | 7) 15 | 8) 17 | 9) none of these | |

4)

```
int sum = 0, mystery;
ifile >> mystery;
while (ifile) {
    sum = sum + mystery;
    ifile >> mystery;
}
cout << "sum = " << sum;
```

- 5)

```
int sum = 0, mystery;
infile >> mystery;
while (mystery > 0) {
    sum = sum + mystery;
    infile >> mystery;
}
cout << "sum = " << sum;
```
- 6)

```
int sum = 0, count, mystery, value;
infile >> mystery;
for (count = 1; count <= mystery; count++) {
    infile >> value;
    sum = sum + value;
}
cout << "sum = " << sum;
```

For the next two questions, consider executing the following code fragment (assume any additional declarations, etc, needed to make the code syntactically correct):

```
int    j = 5;
while (j != 28) {
    j = j + 3;
    cout << j << endl;
}
```

- 7) Exactly how many times will the body of the loop be executed?
- 1) 6 2) 7 3) 8 4) 9 5) none of these
- 8) What is the fourth value printed?
- 1) 5 2) 8 3) 11 4) 17 5) 20
6) 29 7) none of these

-
- 9) In C++, a function prototype is
- 1) a declaration but not a definition. 2) a definition but not a declaration.
3) both a declaration and a definition. 4) neither a declaration nor a definition.

-
- 10) A function SomeFunc has two parameters, alpha and beta, of type int. The data flow for alpha is one-way, into the function. The data flow for beta is two-way, into and out of the function. What is the most appropriate function heading for SomeFunc?
- 1) void SomeFunc (int alpha, int beta)
2) void SomeFunc (int& alpha, int beta)
3) void SomeFunc (int alpha, int& beta)
4) void SomeFunc (int& alpha, int& beta)
-

Assume the following declarations:

```
void Fix(int& , float& );  
int  someInt = 42;  
float someFloat = 3.14;
```

11) Which of the following is an appropriate call of the function `Fix`?

- | | |
|--|---------------------|
| 1) <code>Fix(24, 6.85);</code> | 6) all of the above |
| 2) <code>Fix(someInt, 6.85);</code> | 7) 1 through 4 only |
| 3) <code>Fix(24, someFloat);</code> | 8) 1 and 3 only |
| 4) <code>Fix(someInt, someFloat);</code> | 9) 2 and 4 only |
| 5) <code>Fix(someInt + 5, someFloat);</code> | 10) none of these |

Given the function:

```
void SomeFunc( ... )  
{  
    float alpha;  
    ...  
}
```

12) Which of the following statements about `alpha` is *false*?

- | | |
|--|------------|
| 1) The memory allocated to <code>alpha</code> is deallocated when the function returns. | 5) 1 and 2 |
| 2) A parameter in the function heading can also be named <code>alpha</code> . | 6) 1 and 3 |
| 3) The value of <code>alpha</code> is undefined at the moment control enters the function. | 7) 2 and 3 |
| 4) <code>alpha</code> cannot be accessed directly from code outside the function. | 8) 2 and 4 |

Given the function definition:

```
bool IsZip (float someFloat )  
{  
    return (someFloat == 0.0);  
}
```

13) What is the value of the expression `IsZip(2.4)` ?

- | | | | |
|--------------------------------------|------------------|--------|----------|
| 1) 0.0 | 2) TRUE | 3) 2.4 | 4) FALSE |
| 5) " <code>someFloat == 0.0</code> " | 6) none of these | | |
-

For the next three questions, consider execution of the following program:

```
void DoThis(int& Alpha, int Beta);  
int Temp = 12;  
void main() {  
    int Ben = 5, Jerry = 7;  
    DoThis(Ben, Jerry);  
    cout << "Ben    = " << Ben << endl;  
    cout << "Jerry = " << Jerry << endl;  
    cout << "Temp  = " << Temp << endl;  
}  
  
void DoThis(int& Alpha, int Beta) {  
    Alpha = Alpha + 10;  
    Temp  = Beta;  
    Beta  = 11;  
    Temp  = 14;  
}
```

14) What value is printed for the variable Ben?

- 1) 5 2) 7 3) 11 4) 12 5) 14
6) 15 7) None of these

15) What value is printed for the variable Jerry?

- 1) 5 2) 7 3) 11 4) 12 5) 14
6) 15 7) None of these

16) What value is printed for the variable Temp?

- 1) 5 2) 7 3) 11 4) 12 5) 14
6) 15 7) None of these

17) What is the output of the following code fragment? (All variables are of type `int`.)

```
limit = 8;  
cout << 'H';  
for (loopCount = 10; loopCount <= limit; loopCount++)  
    cout << 'E';  
cout << "LP";
```

- 1) HLP 2) HELP 3) HEELP 4) HEEELP 5) None of these

18) What is the value of sum after execution of the following code? (All variables are of type `int`.)

```
sum = 0;  
for (counter = 2; counter <= 5; counter++)  
    sum = sum + 2 * counter;
```

- 1) 28 2) 18 3) 10 4) 30 5) None of these
-

19) Given the array declaration below, what is the range of valid index values for myArray[]?

```
Int  MaxStuff = 40;
char myArray[MaxStuff + 10];
```

- | | |
|-----------------|------------------|
| 1) 0 through 39 | 5) 1 through 50 |
| 2) 0 through 40 | 6) 1 through 51 |
| 3) 0 through 49 | 7) none of these |
| 4) 0 through 50 | |

20) What is the output of the following program?

```
#include <iostream.h>
void main() {
    int Ray[6] = {100, 200, 300, 400, 500, 600};
    int k;
    for (k = 4; k >= 0; k = k - 2)
        cout << Ray[k] << ' ';
}
```

- | | |
|----------------------------|------------------|
| 1) 600 500 400 300 200 100 | 5) 600 400 200 |
| 2) 400 300 200 100 | 6) 500 300 100 |
| 3) 500 400 300 200 100 | 7) 200 400 600 |
| 4) 100 300 500 | 8) none of these |

21) Given an integer array Fred[] holding 5000 values, which of the code fragments below could be used to print out the values of Fred[1], Fred[3], Fred[5],...?

- | | |
|---|---------------------|
| 1) for (int i = 0; i < 5000; i = i + 2)
cout << Fred[i] << endl; | 6) all of the above |
| 2) for (int i = 1; i < 5000; i = i + 2)
cout << Fred[i] << endl; | 7) 1 and 3 only |
| 3) for (int i = 0; i < 2500; i++)
cout << Fred[2*i] << endl; | 8) 2 and 5 only |
| 4) for (int i = 0; i < 2500; i++)
cout << 2*Fred[i] + 1 << endl; | 9) 1, 3, and 5 only |
| 5) for (int i = 0; i < 2500; i++)
cout << Fred[2*i+1] << endl; | 10) none of these |
-

22) Which of the following statements about C++ arrays is *true*?

- 1) Array components cannot be of floating point types.
 - 2) The index type of an array can be any data type.
 - 3) An array component can be treated as a simple variable of its component type.
 - 4) 1 and 2 above
 - 5) 1, 2, and 3 above
-

23) Given the declaration: `char myName[5] = "Jake";`

which of the following would output "Jake" (without the quotes, of course)?

- | | |
|---|----------------------------|
| 1) <code>cout << myName;</code> | 4) none of the above would |
| 2) <code>for (int ch = 0; ch < 4; ch++)
 cout << myName[ch];</code> | 5) 1 and 2 would |
| 3) <code>for (int i = 0; i < 4; i++)
 cout << myName;</code> | 6) 2 and 3 would |
| | 7) all of the above would |
-

24) Given the declaration

`char table[7][9];`

which of the following stores the character 'B' into the fifth row and second column of the array?

- | | |
|------------------------------------|------------------------------------|
| 1) <code>table[4][1] = 'B';</code> | 2) <code>table[1][4] = 'B';</code> |
| 3) <code>table[5][2] = 'B';</code> | 4) <code>table[2][5] = 'B';</code> |
| 5) <code>table[5] = 'B';</code> | 6) None of these |
-

For the next two questions, consider writing a program that contains the following variable declarations.

```
const int MaxSize = 5;  
int aRay[MaxSize] = {1, 2, 3, 4, 5},  
    bRay[MaxSize] = {27, 13, 4, 4, 2},  
    RayStevens[MaxSize-1] = {0, 0, 0, 0};
```

Indicate whether the proposed statement given in each question is:

- 1) syntactically illegal; i.e., there would be a compile-time error
- 2) syntactically legal, but logically incorrect
- 3) syntactically legal and logically correct, as far as we can tell

25) `aRay = bRay;`

26) `bRay[3] = RayStevens[aRay[2]];`

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

```
// ReverseValues() takes an array of integers and the number of values
// it contains, and reverses the order of the values in the array,
// returning the number of swaps that are performed.
//
// Parameters:
//     List[]      array of integers
//     HowMany     number of values stored in List[]
// Returns:
//     the number of elements that are swapped in reversing List[]
//
int ReverseValues(_____ List[], _____ HowMany) { // line A

    int numSwaps = 0;           // line B
    int tempInt;                // line C
    int Lo = 0;                 // line D
    int Hi = _____;       // line E

    while (Lo < Hi) {           // line F
        tempInt = List[_____]; // line G
        List[Lo] = List[Hi];    // line H
        List[Hi] = _____;  // line I
        numSwaps++;             // line J
        Lo++;                   // line K
        Hi--;                   // line L
    }
    return numSwaps;           // line M
}
```

27) How should the blank preceding the first parameter in line A be filled?

- | | | |
|----------------------------|------------------|---------|
| 1) const int | 2) int | 3) int& |
| 4) it should be left blank | 5) none of these | |

28) How should the blank in line E (initializing Hi) be filled?

- | | |
|------------------|----------------|
| 1) 0 | 2) HowMany |
| 3) HowMany - 1 | 4) HowMany + 1 |
| 5) none of these | |

29) How should the blank in line G be filled?

- | | |
|------------------|-----------|
| 1) Lo | 2) Hi |
| 3) Lo - 1 | 4) Hi + 1 |
| 5) none of these | |

30) How should the blank in line I be filled?

- | | |
|------------------|-----------------|
| 1) List[Lo] | 2) List[Hi - 1] |
| 3) HowMany | 4) 0 |
| 5) none of these | |
-