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 the five-digit course number in the spaces 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.
- Be careful to distinguish integer values from floating point (real) values (containing a decimal point). 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.
- 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.

__________________________
signature
For the next two questions, consider the execution of the following code fragment:

```cpp
int I = 12, J = 0;
while (I > 8){
    I = I - 1;
    J = J + I; //Note: I is not the integer 1
}
cout << "I = " << I << endl;
cout << "J = " << J << endl;
```

1) What value is printed for the variable I:
   1) 4  2) 5  3) 6  4) 7  5) 8  6) 9  7) 10  8) none of the above

2) What value is printed for the variable J:
   1) 0  2) 4  3) 11  4) 21  5) 30  6) 38  7) 45  8) none of the above

3) Which of the following are iteration mechanisms in C++?
   1) while  2) switch  3) for  4) if..else  5) do..while  6) all of the above
   7) 1 and 3 only  8) 2 and 4 only  9) 1, 3 and 5 only  10) none of the above

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

```cpp
int j = 2;
while (j != 28) {
    cout << j << endl;
    j = j + 3;
}
```

4) Exactly how many times will the body of the loop be executed?
   1) 8  2) 9  3) 10  4) 11  5) none of the above

5) What is the eighth value printed?
   1) 2  2) 20  3) 23  4) 26  5) 28  6) 29  7) 47  8) none of these
For the next three questions, assume the input file stream `ifile` is connected to a file containing the following data:

```
5  3  -2  4  1  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) 5  
3) 6  
4) 8  
5) 10  
6) 11  
7) 13  
8) 18  
9) none of the above

6) ```
   int sum = 0, count, mystery, value;
   ifile >> mystery;
   for (count = 1; count <= mystery; count++) {
      ifile >> value;
      sum = sum + value;
   }
   cout << "sum=" << sum;
```  

7) ```
   int sum = 0, mystery;
   ifile >> mystery;
   while (ifile) {
      sum = sum + mystery;
      ifile >> mystery;
   }
   cout << "sum=" << sum;
```  

8) ```
   int sum = 0, mystery;
   ifile >> mystery;
   while (mystery > 0) {
      sum = sum + mystery;
      ifile >> mystery;
   }
   cout << "sum=" << sum;
```  

Consider executing the following program:

```c
void main() {
   int i, j, sum;
   sum = 0;
   for (i = 1; i <= 3; i++) {
      for (j = 1; j <= 3; j++)
         sum++;
      cout << sum << endl;
   }
}
```

9) What is the value printed on the second line of output?

1) 1  
2) 2  
3) 3  
4) 4  
5) 5  
6) 6  
7) 0  
8) none of the above
Now, consider executing the following slightly different program:

```c
void main() {
    int i, j, sum;
    for (i = 1; i <= 3; i++) {
        sum = 0;
        for (j = 1; j <= 3; j++)
            sum++;
        cout << sum << endl;
    }
}
```

10) What is the value printed on the second line of output?

   1) 1  2) 2  3) 3  4) 4  5) 5  6) 6  7) 0  8) none of the above

Assume the following declarations:

```c
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) Fix(24, 6.85);  6) all of the above
   2) Fix(someInt, 6.85);  7) 1 through 5 only
   3) Fix(24, someFloat);  8) 1 and 3 only
   4) Fix(someInt, someFloat);  9) 2 and 4 only
   5) Fix(someInt + 5, someFloat);  10) none of the above

12) A function parameter should be passed by reference if the parameter’s data flow (communication) is:

   1) one-way, into the function  5) 1 and 2 only
   2) one-way, out of the function  6) 1 and 3 only
   3) two-way, into and out of the function  7) 2 and 3 only
   4) all of the above  8) none of the above

13) A function, `someFunc`, has two formal parameters, `Alpha` and `Beta`, both of type `int`. The data flow (communication) for `Alpha` is one-way, into the function. The data flow for `Beta` is two-way, into and out of the function. Which of the following is the most appropriate prototype for `someFunc`?

   1) `someFunc(int Alpha, int Beta);`  4) `someFunc(int& Alpha, int Beta);`
   2) `someFunc(int& Alpha, int Beta);`  5) all of the above
   3) `someFunc(int Alpha, int& Beta);`  6) none of the above
For the next three questions, consider execution of the following program:

```c++
void DoThis(int& Alpha, int Beta) {
    int Temp;
    Alpha = Alpha + 100;
    Temp = Beta;
    Beta = 999;
    Temp = 12;
}

void main() {
    int Temp = 15;
    int Ben = -5, Jerry = 42;
    DoThis(Ben, Jerry);
    cout << “Ben = ” << Ben << endl;
    cout << “Jerry=” << Jerry << endl;
    cout << “Temp = ” << Temp << endl;
}
```

14) What value is printed for the variable Ben?

1) -5  
2) 12  
3) 42  
4) 95  
5) 142  
6) 999  
7) None of the above

15) What value is printed for the variable Jerry?

1) -5  
2) 12  
3) 42  
4) 95  
5) 142  
6) 999  
7) None of the above

16) What value is printed for the variable Temp?

1) -5  
2) 12  
3) 42  
4) 95  
5) 142  
6) 999  
7) None of the above

17) What is the most appropriate function prototype for a function that receives a character letter grade and returns its integer equivalent on a four-point grading scale?

1) void IntEquiv(char );  
2) void IntEquiv(char& );  
3) void IntEquiv(int );  
4) void IntEquiv(int& );  
5) int IntEquiv(char );  
6) int IntEquiv(char& );  
7) all of the above  
8) 1 and 5 only  
9) 5 and 6 only  
10) none of the above

18) What is the output of the following code fragment?

```c++
int Limit = 10;
cout << ‘H’;
for (int LoopCount = 8; LoopCount < Limit; LoopCount++)
    cout << ‘E’;
cout << “LP”;
```

1) HLP  
2) HELP  
3) HEELP  
4) HEEELP  
5) None of the above
19) What loop condition should replace "? ? ?" in the code below in order to produce the output: 8 17 35 71?

```cpp
int N = 8;
do {
    cout << setw(3) << N;
    N = N*2 + 1;
} while (? ? ?);
```

- 1) N <= 71
- 2) N < 71
- 3) N <= 35
- 4) N < 35
- 5) N > 143
- 6) none of the above

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

```cpp
int MaxStuff = 65;
char myArray[MaxStuff + 10];
```

- 1) 0 through 75
- 2) 0 through 74
- 3) 0 through 76
- 4) 1 through 75
- 5) 1 through 74
- 6) 1 through 76
- 7) none of the above

21) What is the output of the following program?

```cpp
void main() {
    int Ray[5] = {100, 200, 300, 400, 500};
    int k;
    for (k = 4; k >= 1; k--)
        cout << Ray[k] << ' ';
}
```

- 1) 100 200 300 400 500
- 2) 400 300 200 100 0
- 3) 500 400 300 200 100
- 4) 4 3 2 1 0
- 5) 500 400 300 200
- 6) 400 300 200 100
- 7) 100 200 300 400
- 8) none of the above
22) Given an integer array Fred[] holding 5000 values, which of the code fragments below could be used to print out the values of Fred[0], Fred[2], Fred[4], etc:

1) for (int i = 0; i < 5000; i = i + 2)
   cout << Fred[i] << endl;

2) for (int i = 0; i < 2500; i++)
   cout << Fred[i] << endl;

3) for (int i = 0; i < 2500; i++)
   cout << Fred[2*i] << endl;

4) for (int i = 0; i < 2500; i++)
   cout << 2*Fred[i] << endl;

5) for (int i = 0; i < 5000; i++)
   cout << Fred[i] << endl;

6) all of the above

7) 1 and 2 only

8) 1 and 3 only

9) 1, 2, and 3 only

10) none of the above

For the next three questions, consider writing a program that contains the following variable declarations and function prototypes.

\[
\text{\begin{verbatim}
const int MaxSize = 5;
void FillArray(int Ray[], int arrayDimension);
int aRay[MaxSize] = {1, 2, 3, 4, 5},
                 bRay[MaxSize] = {27, 13, 4, 4, 2},
                 RayStevens[MaxSize-1] = {0, 0, 0};
\end{verbatim}}
\]

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

23) aRay[MaxSize] = bRay[MaxSize];

24) FillArray(4, aRay);

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

```
// MaxValue() takes an array of integers and returns the index of the
// largest value in the array.
//
// Parameters:
// List[] array of integers
// HowMany number of values stored in List[]
// Returns:
// the index of the largest value in List[]
//
int MaxValue(List[], HowMany) { / / line A
    int maxSoFar = / / line B
    int Look; // line C
    for (Look = 1; / / line D
        if (List[maxSoFar] < List[Look]) // line E
        / / line F
    return maxSoFar; // line G
}
```

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

1) int  2) const int  3) int&
4) const int&  5) none of the above

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

1) int  2) int&  3) int& const
4) none of the above

28) How should the blank in line B be filled? Be careful.

1) 0  2) -1  3) List[0]
4) HowMany  5) none of the above

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

1) Look <= HowMany  2) Look < HowMany  3) Look < maxSoFar
4) Look == HowMany  5) none of the above

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

1) Look = maxSoFar  2) Look == maxSoFar  3) maxSoFar = Look
4) maxSoFar = List[Look]  5) none of the above