



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 in the space provided below.
- Print your name and ID number on the Opscan form; be sure to code your ID number correctly on the Opscan form. Code **Form A** on the Opscan; code your section **group** number:

Struble	Engel 223	1
Henry	Smyth 146	2
Tucker	Litton-Reeves 1670	3
McQuain	Litton-Reeves 1870	4
	Engel 223	5

- 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, assume that it is. 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 (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 finished, 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 equal-valued 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 (Last, First) _____ printed

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:

```
int I, J = 1;
for (I = 5; I >= 2; I--){
    J = J - I;
}
cout << "I = " << I << endl;
cout << "J = " << J << endl;
```

1) What value is printed for the variable I:

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

2) What value is printed for the variable J:

- 1) 14 2) 6 3) 15 4) -12 5) -13
6) -14 7) 6 8) none of the above

Consider executing the following code fragment (assume any additional declarations, etc, needed to make the code syntactically correct):

```
int j = 3;
double varq = 1.5, newq = 0.0;
while (j < 47) {
    newq = newq + varq;
    j = j + 4;
}
```

3) How many times will the body of the loop be executed?

- 1) 11 2) 12 3) 13 4) 46 5) 47
6) none of the above

4) What is the value of newq after the last iteration of the loop?

- 1) 1.5 2) 16.5 3) 18.0 4) 19.5 5) 44.5
6) 269.5 7) none of the above

5) A function, someFunc, has two formal parameters, A1 of type int and B2, both of type string. The data flow (communication) for variable A1 is two-way, into and out of the function. The data flow for variable B2 is one-way, into the function. Which of the following is the **most appropriate** prototype for someFunc?

- 1) void someFunc(int A1, string B2);
2) void someFunc(int& A1, string B2);
3) void someFunc(int A1, const string& B2);
4) void someFunc(int& A1, const string& B2);
5) all of the above
6) none of the above

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

2 3 -5 5 1 -4

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) 2 2) 6 3) -2 4) 4 5) 5
 6) -5 7) 0 8) 18 9) none of the above

6)

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

7)

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

8)

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

Consider executing the following code fragment (assume any additional declarations, etc, needed to make the code syntactically correct):

```
int j = 2;
while (j != 64) {
    j = j * 2;
    cout << j << endl;
}
```

9) How many times will the body of the loop be executed?

- 1) 4 2) 5 3) 6 4) infinite loop 5) none of the above

10) What is the last value printed?

- 1) 64 2) 32 3) 16 4) 19
 5) 17 6) 13 7) 10 8) none of these

Consider executing the following program:

```
void main() {
    int i, j, A[2][4] = {1,2,3,4,5,6,7,8};
    for (i = 0; i < 3; i++)
        for (j = 0; j < 1; j++)
            cout << A[j][i] << endl;
}
```

11) What is the value printed on the third line of output?

- 1) 1 2) 2 3) 3 4) 4 5) 5 6) 6
7) 7 8) 8 9) none of the above
-

Consider executing the following program:

```
void main() {
    int i, j, sum;
    sum = 0;
    for (i = 0; i < 6; i=i+2) {
        for (j = 0; j < 4; j=j+2)
            sum = sum + i + j;
        cout << sum << endl;
    }
}
```

12) What is the value printed on the last line of output?

- 1) 0 2) 1 3) 2 4) 16 5) 18 6) 20
7) 42 8) none of the above
-

Assume the following declarations:

```
void Fix(double realvar, int& intvar );
int someInt = 12;
double someFloat = 6.28;
```

13) Which of the following would represent an appropriate call(s) of the function Fix?

- 1) Fix(6.85, 24);
2) Fix(6.85, someInt);
3) Fix(someFloat, 24);
4) Fix(someFloat, someInt+5);
5) Fix(someFloat+2.0, someInt);
6) all of the above
7) 1 and 2 only
8) 2 and 5 only
9) 2, 4 and 5 only
10) none of the above
-

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

```

void DoThis(int Ben, int& Jer);
int Tmp = 15;
void main() {
    int Alpha = -5, Beta = 42;
    DoThis(Alpha, Beta);
    cout << "Alpha = " << Alpha << endl;
    cout << "Beta = " << Beta << endl;
    cout << "Tmp = " << Tmp << endl;
}

void DoThis(int Ben, int& Jer) {
    int Tmp;
    Tmp = Ben;
    Ben = Ben + 100;
    Jer = Tmp;
}

```

14) What value is printed for the variable Alpha?

- 1) -5 2) 15 3) 42 4) 95 5) 142
 6) 999 8) None of the above

15) What value is printed for the variable Beta?

- 1) -5 2) 15 3) 42 4) 95 5) 142
 6) 999 8) None of the above

16) What value is printed for the variable Tmp?

- 1) -5 2) 15 3) 42 4) 95 5) 142
 6) 999 8) None of the above

17) Given the declaration

```
char table[7][9];
```

which of the following stores the character 'B' into the sixth column and third row of the array?

- 1) table[3][6] = 'B'; 2) table[6][3] = 'B';
 3) table[2][5] = 'B'; 4) table[5][2] = 'B';
 5) table[4][7] = 'B'; 6) table[7][4] = 'B';
 7) table[3] = 'B'; 8) None of these

18) Given the function heading

```
void GetNums( int    howMany,
              float& alpha,
              float& beta )
```

which of the following is a valid function prototype for GetNums?

- 1) void GetNums(int howMany, float& alpha, float& beta);
 2) void GetNums(int, float&, float&);
 3) int GetNums(int, float&, float&);
 4) 1 and 2 above
 5) 1, 2, and 3 above
 6) none of the above

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

```
// CalcAverage takes an array of grades stored as integers
// and the number of values it contains, prints the grades
// and determines the average grade.
//
// Parameters:
//   Grades[]      array containing grades.
//   NumOfGrades  number of values stored in Grades[]
//
void CalcAverage(_____ Grades[], _____ NumOfGrades) // Line A
{
    int Sum = 0;
    int LoopCounter;

    for(LoopCounter=0;_____;LoopCounter++) // Line B
    {
        Sum=Sum+_____; //Line C
    }

    cout<<"The Grades are:"<<endl;
    for(LoopCounter=0;_____;LoopCounter++) //Line D
        cout<<Grades[LoopCounter]<<'\\t';

    cout<<"Average = "<<_____; //Line E
}
```

19) 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 | |

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

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

21) How should the blank in lines B & D be filled?

- | | |
|-----------------------|-------------------------------|
| 1) Sum < NumOfGrades | 2) LoopCounter < NumOfGrades |
| 3) Sum <= NumOfGrades | 4) LoopCounter <= NumOfGrades |
| 5) none of these | |

22) How should the blank in line C be filled?

- | | |
|--------------------|------------------------|
| 1) Sum/NumOfGrades | 2) Grades[NumOfGrades] |
| 3) Sum | 4) Grades[LoopCounter] |
| 5) Sum/LoopCounter | |

23) What is the output of the following program?

```
void main() {
    int Ray[5] = {100,200,300,400,500};
    int k;
    for (k = 1; k < 6; k=k+2)
        cout << Ray[k-1] << ' ';
}
```

- 1) 100 200 300 400 500
- 2) 100 200 300 400
- 3) 500 400 300 200 100
- 4) 1 3 5
- 5) 1 2 3 4 5
- 6) 200 300 400
- 7) 100 300 500
- 8) none of the above

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

```
const int MAXSTUFF = 20;
char myArray[MAXSTUFF * 2];
```

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

Assume the following declarations for a program to keep track of a sporting event schedule:

```
const int NUMGAMES = 5;
const int NUMTEAMS = 5;
const string TEAMS[_____] = { "Virginia Tech", "Miami", "Virginia",
                               "Syracuse", "West Virginia" };
string HomeTeam[_____] ;
string AwayTeam[_____] ;
string Winner[_____] ;
```

25) What most appropriately fills in the blank in the declaration of TEAMS?

- 1) NUMGAMES 2) NUMTEAMS 3) 5 4) NUMGAMES + 1 5) NUMTEAMS + 1
- 6) none of the above

26) What most appropriately fills in the blank in the declaration of Winner?

- 1) NUMGAMES 2) NUMTEAMS 3) 5 4) NUMGAMES + 1 5) NUMTEAMS + 1
- 6) none of the above

27) The arrays HomeTeam, AwayTeam, and Winner are

- 1) parallel arrays 2) single dimensional arrays 3) multi-dimensional arrays
- 4) all of the above 5) 1 and 2 only 6) 1 and 3 only
- 7) none of the above

Assume that the following array is declared as follows:

```
int aMaze[MAZEHEIGHT][MAZEWIDTH];
```

28) Suppose a function named `FindWay` exists and accepts a maze as a one-way parameter into the function and returns no values. The most appropriate function prototype is:

- 1) `void FindWay(int Maze[][MAZEWIDTH], int height, int width);`
- 2) `void FindWay(const int Maze[][MAZEWIDTH], int height, int width);`
- 3) `void FindWay(int Maze[MAZEHEIGHT][], int height, int width);`
- 4) `void FindWay(const int Maze[MAZEHEIGHT][], int height, int width);`
- 5) `void FindWay(int Maze[MAZEHEIGHT][MAZEWIDTH], int height, int width);`
- 6) all of the above
- 7) none of the above

29) The maximum number of elements that can be stored in `aMaze` is

- | | |
|---|-------------------------------------|
| 1) $(MAZEHEIGHT - 1) * (MAZEWIDTH - 1)$ | 2) $(MAZEHEIGHT - 1) * MAZEWIDTH$ |
| 3) $MAZEHEIGHT * (MAZEWIDTH - 1)$ | 4) $MAZEHEIGHT * MAZEWIDTH$ |
| 5) $(MAZEHEIGHT + 1) * (MAZEWIDTH + 1)$ | 6) $(MAZEHEIGHT + 1) * MAZEWIDTH$ |
| 7) $MAZEHEIGHT * (MAZEWIDTH + 1)$ | 8) not enough information available |
| 9) none of the above | |

30) Suppose the first few lines of a function are as follows:

```
void Calc( float beta )  
{  
    alpha = 3.8 * beta;
```

Assuming the program compiles, the variable `alpha` is

- 1) a local variable
- 2) a global variable
- 3) a formal parameter
- 4) an actual parameter
- 5) none of the above