

CSCI 261 EXAM 1 REVIEW

July 21, 2010

Identify each term that is a valid C++ identifier, or explain why it is not.

A oxygen%

B The_Hot_Tomatoes

C double

D 2TooMany

E Ozone

Name each part of the of C++ declarations below.

```
int theCounter;
```

```
string theWordStop("Stop"), strData;
```

```
const double PI(acos(-1));
```

Consider the snippet of code at the right. What are the values of `x`, `y`, and `z` after the variables have been initialized?

```
1 double x( 3/4 );  
2 double y( 3%4 );  
3 double z( 3.0/4 );
```

Write the English statement as a C++ Boolean expression (a test condition that might be used in an `if()` or `while()` statement).

- A The real value x is between 1.5 (exclusive) and 2 (inclusive).
- B The integer z is greater than zero and divides the integer y evenly.
- C m is less than zero, or m is greater than or equal to 4.

What does the following snippet print to the screen?

```
1 #include <iostream>
2 using namespace std;
3
4 int main( void )
5 {
6     int x( 3 + 4 % 6 );
7
8     if( x <= 4 ) {
9         cout << "Treat_your_password_like_your_toothbrush." << endl;
10    } else if( x == 5 ) {
11        cout << "Don't_let_anybody_else_use_it," << endl;
12    } else if( x == 6 ) {
13        cout << "and_get_a_new_one" << endl;
14    } else if( x >= 7 ) {
15        if( x == 7 ) {
16            cout << "every_six_months." << endl;
17        } else {
18            cout << "--_Clifford_Stoll" << endl;
19        }
20    } else {
21        cout << "Author_of_'The_Cuckoo's_Egg'" << endl;
22    }
23    return 0;
24 }
```

What does the following snippet print to the screen?

```
1 #include <iostream>
2 using namespace std;
3
4 int main( void )
5 {
6     int x( 3 + 4 % 6 );
7
8     if( x <= 4 ) {
9         cout << "Treat_your_password_like_your_toothbrush." << endl;
10    } else if( x == 5 ) {
11        cout << "Don't_let_anybody_else_use_it," << endl;
12    } else if( x == 6 ) {
13        cout << "and_get_a_new_one" << endl;
14    } else if( x >= 7 ) {
15        if( x == 7 ) {
16            cout << "every_six_months." << endl;
17        } else {
18            cout << "--_Clifford_Stoll" << endl;
19        }
20    } else {
21        cout << "Author_of_'The_Cuckoo's_Egg'" << endl;
22    }
23    return 0;
24 }
```

Suppose that `x` is not initialized, and has a garbage value. Which `cout` statement will **never** print to the screen?

What type of looping control structure should be used if a sequence of statements must execute for a pre-determined number of times?

Rewrite the code snippet at the right using a for loop.

Declare the counter variable such that its lifetime and scope is limited to the for loop.

```
2 int counter(3);  
3 int value(10);  
4 while( counter >= 0 ) {  
5     value *= ++value * --counter;  
6 }
```

Examine the code snippet at the right.

- A In total, how many '*' characters will be printed?
- B How many '+' characters will be printed on the first iteration of the outermost loop (m=1)?
- C How many '+' characters will be printed on the last iteration of the outermost for loop?
- D How many complete lines will be displayed to the user?

```
1 for( int m=1 ; m <= 3; m++ ) {  
2     cout << '*';  
3     for( int j=m-1; j>0; j-- ) {  
4         cout << '+';  
5     }  
6     cout << endl;  
7 }
```

Suppose a user types the following sequence of characters at the keyboard (there is a single whitespace character, and the final character is period):

324_0.123e-10.027e-10c4.

And the keyboard is read by a program running the code sequence to the right.

```

35 int i1, i2, i3;
36 char c1, c2, c3, c4, c5, c6;
37 double d1, d2, d3;
38
39 cin >> i1;
40 cin >> c1;
41 cin >> d1;
42 cin >> c2;
43 cin >> i2;
44 cin >> c3;
45 cin >> d2;
46 cin >> c4;
47 cin >> c5;
48 cin >> i3;
49 cin >> c6;
50 cin >> d3;

```

Complete the table below: provide the value of all variables that are successfully read by `cin` before it encounters a failure state. Note that the variable order in the table is identical to the order of input operations in the code snippet.

Variable	i1	c1	d1	c2	i2	c3
Value						
Variable	d2	c4	c5	i3	c6	d3
Value						

In a sentence, explain why `cin` entered a failure state.

Snippet

```

1 // Open file
2 ifstream infile( "DATA" );
3
4 int N; // number of datapoints
5 // Read N, the number of datapoints in file
6 infile >> N;
7
8 // Reading loop
9 int value; // values read from file
10 for( int n=0; n<N ; n++ ) {
11     infile >> value;
12 }
13
14 // Be sure to close!
15 infile.close();

```

This code snippet reads data from a “counter” based data file (below) The comments are correct, and the snippet compiles and runs correctly. However, the snippet doesn’t contain any logic for detecting when a failure occurs on the input stream.

Example Data

```

48932
8374 839 887
3485 8900 -23873 45
1 2 3
432 546 -874
23232
1857484
4384
:
:

```

This file has 48932 data points in it, so clearly only a portion of the datafile is shown.

Add logic so that:

1. All potential `infile` failures display an appropriate message to the user, then the program calls `exit(1)`;
2. If the failure occurs while reading in a data point (line 11), then *the error message should contain the data point number (not the index) that caused the failure.*

You are allowed to add new variables to the program. You may rewrite the entire snippet or simply rewrite the portions that need modification.

Write a C++ statement that declares an array of 4 Boolean values.

Write a C++ statement that declares an array of 4 Boolean values.

Now do it another way...

Write a C++ statement that declares an array of 4 Boolean values.

Now do it another way...

Write a C++ statement that declares an array of 14 doubles initialized to zero.

Write a C++ statement that declares an array of 4 Boolean values.

Now do it another way...

Write a C++ statement that declares an array of 14 doubles initialized to zero.

Can you write a C++ statement of 1,000,000 Boolean values initialized to true?

Write a **snippet** of C++ that declares an array of 1,000,000 Boolean values and initializes them to true.

Write a **snippet** of C++ that declares a square $2d$ array of 16 char variables initialized to the first 16 capital letters of the alphabet.

finis