Instructions: This homework assignment focuses primarily on some of the basic syntax and semantics of C++. The answers to the following questions can be determined from Chapters 6 and 7 of the lecture notes and Chapters 2 through 6 of the text.

After you have analyzed the questions and decided what answers you believe are correct, you may find it useful to write some short programs to test your logic.

Opscan forms will be passed out in class. Write your name and code your ID number on the opscan form. Turn in your completed opscan at the place and time specified by your Instructor. Opscans will not be accepted at any other place or time.

For questions 1 through 4, consider executing the code fragment:

```cpp
bool A, B, C, D;
// code that assigns values to A, B, C, and D
if (A && B)
    if (!C || !D)
        cout << "one" << endl;
    else if (D)
        cout << "two" << endl;
    else
        cout << "three" << endl;
else if (C != D)
    cout << "four" << endl;
else if (C)
    cout << "five" << endl;
else
    cout << "six" << endl;
```

1) What of the following sets of values for A, B, C, and D would cause the string "one" to be printed?

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>All of the above</td>
<td>2 and 3 only</td>
<td>1 and 2 only</td>
<td>2, 3 and 4 only</td>
<td>None of these</td>
</tr>
</tbody>
</table>

2) What of the following sets of values for A, B, C, and D would cause the string "four" to be printed?

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>All of the above</td>
<td>2 and 3 only</td>
<td>1 and 2 only</td>
<td>1, 3 and 4 only</td>
<td>None of these</td>
</tr>
</tbody>
</table>

3) No matter what values A, B, C and D have, the given code fragment will produce some output.

1) True 2) False 3) None of these
4) What of the following sets of values for A, B, C, and D would cause the string "three" to be printed?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>2)</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>3)</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>4)</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

5) All of the above

6) 1 and 2 only

7) 1, 2 and 3 only

8) 2 and 3 only

9) 1, 3 and 4 only

10) None of these

5) Consider executing the following code fragment (assume x is an int variable):

```
if (x <= 0)
    cout << "One" << endl;
else if (x <= 10)
    cout << "Two" << endl;
else if (x <= 20)
    cout << "Three" << endl;
```

The string "Two" will be printed if and only if x satisfies the condition:

1) x <= 10

2) 0 < x and x < 10

3) 0 < x and x <= 10

4) 0 <= x and x < 10

5) 0 <= x

6) None of these

For questions 6 and 7, consider execution of the following switch statement:

```
int Enter = 10;
cin >> Enter;

switch (Enter) {
    case 1: Enter = -4;
    case 2: Enter = -6;
    case 4: break;
    case 6: Enter = -8;
        break;
    default: Enter = -1;
}
```

6) What would the value of Enter be after execution of this code if the value read for Enter were 4?

1) -4

2) -6

3) -8

4) -1

5) 10

6) None of these

7) What would the value of Enter be after execution of this code if the value read for Enter were 1?

1) -4

2) -6

3) -8

4) -1

5) 10

6) None of these

8) What value is printed for loopCount if the following code fragment is executed?
int loopCount = 1, alpha = 5;
while (loopCount <= 145) {
    alpha = alpha + 7;
    loopCount++;
}
cout << "loopCount = " << loopCount << endl;

1) 1 3) 145 5) None of these
2) 144 4) 146

9) What value is printed for someInt when the following code fragment is executed?

int someInt = 273;
while (someInt > 500)
    someInt = someInt - 3;
cout << "someInt = " << someInt << endl;

1) 270 3) 497 5) infinite loop
2) 273 4) 500 6) None of these

10) What is the logical condition under which the following while loop will terminate?

int Beta = 5;
while (Beta >= 0 && Beta < 10) {
    cout << Beta << endl;
    cin >> Beta;
}

1) Beta < 0 && Beta >= 10 4) Beta <= 0 || Beta >= 10
2) Beta <= 0 && Beta > 10 5) Beta < 0 || Beta >= 10
3) Beta < 0 || Beta > 10 6) None of these

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

int n = 1;
while (n <= 5) {
    cout << n << ' ';
    n++;
}

1) 1 2 3 4 5 3) 1 1 1 forever 5) 2 3 4 5 6
2) 1 2 3 4 4) 2 3 4 5 6) None of these

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

int n = 1;
while (n <= 5) {
    n++;
    cout << n << ' ';
}

1) 1 2 3 4 5 3) 1 1 1 forever 5) 2 3 4 5 6
2) 1 2 3 4 4) 2 3 4 5 6) None of these

13) What is the output of the following code fragment? (Be careful here.)
int n = 1;
while (n <= 5)
    cout << n << ' ';
    n++;

1) 1 2 3 4 5 3) 1 1 1 forever 5) 2 3 4 5 6
2) 1 2 3 4 4) 2 3 4 5 6)

14) When designing a specification for an input file, which of the following would be a poor choice for a sentinel value?

1) a value of 29 for voter ages 6) 1 and 2 only
2) a value of 1025 for SAT scores 7) 1 and 3 only
3) a value of -1 for student heights 8) 2 and 3 only
4) a value of "No one" for student names 9) 2 and 4 only
5) All of them 10) None of these

15) With respect to the loop in the following main() function, what is missing?

#include <iostream>
using namespace std;
int main() {
    int loopCount = 4;
    int Output = 7;
    while (loopCount <= 8) {
        cout << Output << endl;
        Output++;
    }
    return 0;
}

1) the initialization of the loop control variable
2) the testing of the loop control variable
3) the update of the loop control variable
4) Nothing is missing.

16) Indicate where (if at all) the following loop needs a priming read.

int Sum = 0; // line 1
int Number; // line 2
while (inFile) { // line 3
    Sum = Sum + Number; // line 5
    inFile >> Number; // line 6
} // line 7

1) before line 1 4) between lines 5 and 6
2) between lines 2 and 3 5) between lines 6 and 7
3) at line 4 6) No priming read is necessary.
17) To produce the output 2 4 6 8 10, what loop condition should be used in the blank below?

```cpp
int N = 0;
do {
    N = N + 2;
    cout << N << ' ';
} while (_______);
```

1) N <= 10  
2) N < 10  
3) N < 8  
4) N >= 2  
5) N > 8  
6) None of these

18) After execution of the following code, what value is printed for Length?

```cpp
int Length = 5,
    Count = 4;
while (Count <= 6) {
    if (Length >= 100)
        Length = Length - 2;
    else
        Length = Count * Length;
    Count++;
}
```

```cpp
cout << "Length = " << Length << endl;
```

1) 20  
2) 98  
3) 100  
4) 600  
5) None of these

19) In the following code fragment, a semicolon appears at the end of the line containing the while condition.

```cpp
cout << 'A';
int loopCount = 1;
while (loopCount <= 3);
{
    cout << 'B';
    loopCount++;
}
cout << 'C';
```

The result will be:

1) the output AC  
2) the output ABC  
3) the output ABBBC  
4) a compile-time error  
5) an infinite loop  
6) None of these
20) What is the output of the following code fragment?

```cpp
int Sum = 0,
    outerCount = 1;
while (outerCount <= 3) {
    int innerCount = 1;
    while (innerCount <= outerCount) {
        Sum = Sum + innerCount;
        innerCount++;  
    }
    outerCount++;  
}  
cout << Sum << endl;
```

1) 1   3) 10   5) 35
2) 4   4) 20   6) None of these

21) Which `for` loop is equivalent to the following `while` loop? Equivalent here means that the value of each of the variables would be the same when the code has completed execution.

```cpp
int count = -5, sum = 0;
while (count <= 15) {
    sum = sum + count;
    count++;  
}
```

1) `int count, sum = 0;
   for (count = -5; count <= 15; count++)
       sum = sum + count;
   `  
2) `int count, sum = 0;
   for (count = -5; count <= 15; count++) {
       sum = sum + count;
       count++;  
   }
   `  
3) `int count, sum;
   for (count = -5, sum = 0; count <= 15; count++) {
       count++;  
       sum = sum + count;
   }
   `  
4) `int count, sum = 0;
   for (count = 1; count <= 21; count++)
       sum = sum + count;
   `  
5) All of these   8) 1 and 4 only
6) 1 and 2 only   9) None of these
7) 1 and 3 only
22) What is the output of the following code fragment?

```cpp
for (loopCount = 1; loopCount <= 6; loopCount = loopCount + 2)
    cout << loopCount << ' ';
cout << "Done" << endl;
```

1) Done  
2) 1 Done  
3) 1 3 Done  
4) 1 3 5 Done  
5) 1 3 5 7 Done  
6) None of these

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

```cpp
for (loopCount = 1; loopCount > 3; loopCount++)
    cout << loopCount << ' ';
cout << "Done" << endl;
```

1) Done  
2) 1 Done  
3) 1 2 Done  
4) 1 2 3 Done  
5) 1 2 3 4 Done  
6) None of these

24) In the following code fragment, the programmer mistakenly placed a semicolon at the end of the for statement heading. What is the result?

```cpp
cout << 'A';
for (count = 1; count <= 3; count);
    cout << 'B';
    cout << 'C';
```

1) a compile-time error  
2) an infinite loop  
3) the output ABC  
4) the output ABBC  
5) the output ABBBC  
6) None of these

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

```cpp
int n = 2;
for (loopCount = 1; loopCount <= 3; loopCount++) {
    while (n <= 4)
        n = 2 * n;
}
cout << n << endl;
```

1) 4  
2) 8  
3) 16  
4) 32  
5) 64  
6) None of these