1. Are the following legal identifiers in C++? Circle yes or no after each. (15 points)
   a. SumOf15Grades   yes  no
   b. 15_grade_sum     yes  no
   c. _15_grade_sum    yes  no
   d. integer          yes  no
   e. grades!          yes  no
   f. grade:sum:fifteen yes  no
   g. #grade_sum       yes  no
   h. ____              yes  no
   i. int               yes  no
   j. switch            yes  no
   k. branch            yes  no
   l. const             yes  no
   m. constant          yes  no
   n. long              yes  no
   o. character         yes  no

2. What is printed out if the following code is run: (7 points)
   ```
   int x = 33;
   char c = 'G';
   cout << (x++) * 3 << '\t' << --c;
   ```

   99 F

3. Determine the final values of i, j, and k after the following code is run: (9 points)
   ```
   int i = 10, j = 4, k = 5;
   k = i >= j;
   i *= j + 1;
   j /= ++i % 2;
   ```

   i = 51
   j = 4
   k = 1
4. Determine if the expressions below are true, false, or unknown: (30 points)
   a. $3 \geq 3.0$  
      true  false  unknown
   b. -739  
      true  false  unknown
   c. "John" > "john"  
      true  false  unknown
   d. "John" != "John"  
      true  false  unknown
   e. "John" > "Johnny"  
      true  false  unknown
   f. Bob < bob  
      true  false  unknown
   g. false < true  
      true  false  unknown
   h. 0  
      true  false  unknown
   i. '0'  
      true  false  unknown
   j. 'G' < 'X'  
      true  false  unknown
   k. true && (x || y)  
      true  false  unknown
   l. false && (x || y)  
      true  false  unknown
   m. true || (x && y)  
      true  false  unknown
   n. false || (x && y)  
      true  false  unknown
   o. $3 > 0$ || $15 < 0$ || $((197347 > 3793494) && (37337 > 197))$  
      true  false  unknown

5. Given the code below, write a line of code that sets c equal to the character represented by the
   ASCII value stored in x. (6 points)
   ```
   int x = 65;
   char c;

   c = x;  // OR (better) c = static_cast<char>(x);
   ```

6. Explain in one sentence the difference between a while loop and a do/while loop. (6 points)

   A do/while loop is guaranteed to execute at least once because the
   expression is checked at the end of the loop instead of the beginning
   like a while loop.
7. Describe briefly what the following header files are used for: (15 points)
   a. iostream
      primarily used for cin and cout

   b. iomanip
      primarily used to manipulate I/O with things like setw and setprecision

   c. string
      used to be able to create strings (sequences of characters)

   d. fstream
      primarily used for ifstream/ofstream to read/write to files

   e. cmath
      used for a variety of math functions such as pow or sin

8. Explain how the following will alter the output (10 points)
   a. setprecision(n)
      Sets the maximum # of decimal places values should print out to

   b. setw(n)
      Sets the total space (columns = n) for the next thing to be output in. Will fill the extra space (if any) with spaces.

   c. left
      default alignment in the context of setw is to be right aligned and put the padding on the left. Using left first will left align the text and put the padding on the right.
9. The code below does not seem to work as expected: (12 points)

```cpp
string name; // Line 1
int age; // Line 2
cout << "Enter your name: "; // Line 3
cin >> name; // Line 4
cout << "Enter your age: "; // Line 5
cin >> age; // Line 6
cout << name << "'s age is " << age << endl; // Line 7
```

I typed in my name, Joe Smith, but it never let me enter anything for my age. At the end it told me I was 0 years old! Explain what happened and change one line in the above program to fix it. 

`cin` splits input on blank spaces. Replace line 4 with `getline(cin, name);`

10. Give the syntax for, and explain briefly, the two ways to write comments in C++. (5 points)

```cpp
// This is a single line comment
/*/ This is a
    multiline comment */
```

11. Write the following code in a more succinct manner: (5 points)

```cpp
if (p > 0)
    if (q > 0)
        if (r > 0 || s < 0)
            cout << "Success!" << endl;
    if (p > 0 && q > 0 && (r > 0 || s < 0))
        cout << "Success!" << endl;
```

12. Suppose the user enters a positive integer value into `max`. Write a portion of code containing a loop that will print out every power of 2 so long as that value is less than `max`. Use one variable (other than `max`) and do not use the `pow` function. (20 points)

```cpp
int value = 1;
while (value < max)
{
    cout << value << endl;
    value *= 2;
}
```
13. Given the following code: (10 points)

```cpp
switch (ch)
{
    case 'a':
        cout << "You got an A!" << endl;
        break;
    case 'b':
        cout << "You got a B!" << endl;
        break;
    case 'c':
        cout << "You got a C!" << endl;
    default:
        cout << "You did not pass." << endl;
}
```

What will the output be when ch is equal to:

1. a
   You got an A!
2. A
   You did not pass.
3. b
   You got a B!
4. c
   You got a C!
   You did not pass.
5. d
   You did not pass.

14. What are the results of the following expressions (only write them to 3 decimal places): (5 points)

   a. 8.0 / 3.0 = 2.666
   b. 8.0 / 3 = 2.666
   c. 8 / 3.0 = 2.666
   d. 8 / 3 = 2
15. On the following page, write an entire syntactically correct program (no comments required) that will do the following: (85 points)
   a. Open a file called data and read in 5 values separated by spaces:
      i. the first name of the employee (a string value)
      ii. the number of hours worked (an integer value)
      iii. the hourly pay rate (a floating-point value)
      iv. a yearly bonus amount (a floating-point value)
      v. a boolean value of 0 or 1, where 1 indicates that they definitely get their bonus and 0 indicates they may not get their bonus (see below).
   b. The pay for the person is calculated as \textit{hours worked} \times \textit{hourly pay rate} + \textit{yearly bonus}, but they only get the yearly bonus if the boolean value above indicates they get their bonus or if the number of hours worked is over 2,000.
   c. Write to a file called results the following (where \textit{name} is given from the first input value and \textit{xxx.xx} is to two decimal place accuracy and calculated as described above):
      \begin{align*}
      \text{Pay for } \textit{name} & \text{ is } \textit{xxx.xx} \\
      \end{align*}

\textbf{SCRATCH WORK IS FINE HERE, BUT WILL NOT BE GRADED. WRITE PROGRAM ON NEXT PAGE.}
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;

int main()
{
    ifstream input;
    input.open("data");

    string name;
    int hours;
    double rate, bonus;
    bool got_bonus;

    input >> name >> hours >> rate >> bonus >> got_bonus;
    input.close();

    ofstream output;
    output.open("results");

    output << "Pay for " << name << " is $" <<
            fixed << setprecision(2);

    if (got_bonus || hours > 2000)
        output << hours * rate + bonus << endl;
    else
        output << hours * rate << endl;

    output.close();

    return (0);
}
16. (Extra credit - 16 points) In C++ there are bitwise operators. Bitwise operators perform operations on each bit of a value. For example, the decimal value for 20 is 10100 and the decimal value for 8 is 01000. If I were to complete a bitwise OR operation on the two values, the result would be calculated as follows:

First digit (leftmost) = 1 OR 0 = 1
Second digit = 0 OR 1 = 1
Third digit = 1 OR 0 = 1
Fourth digit = 0 OR 0 = 0
Fifth digit = 0 OR 0 = 0

Therefore, 10100 bitwise OR 01000 = 11100

There is a bitwise operator called XOR, using the symbol ^, with truth table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A ^ B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Assume \(a\) and \(b\) are integers. The following operations are then conducted:

\[a = a ^ b;\]
\[b = a ^ b;\]
\[a = a ^ b;\]

Describe the result of the above operations in one sentence.

The two variables are swapped.