1. Assume a string object has been defined as follows:  
   string description;
   1. Write a cin statement that reads in a one-word description  
      **cin >> description;**
   2. Write a statement that reads in a description that can contain multiple words separated by blanks.  
      **getline (cin, description);**
2. Write a definition statement for a character array large enough to hold any of the follow strings:
   1. “Billy Bob’s Pizza”
   2. “Downtown Auto Supplies”
   3. “Betty Smith School of Architecture”
   4. “ABC Cabinet Company”  
      **char letter [31];**
3. Assume the array name is defined as follows:  
   char name [25];
   1. Using a stream manipulator, write a cin statement that will read a string into a name, but will read no more characters than name can hold.  
      **cin >> setw(25) >> name;**
   2. Using the getline functions, write a cin statement that will read a string into a name that will read no more characters than name can hold.  
      **getline (cin, name[25]);**
4. Assume the following variables are defined:  
   int age;  
   double pay;  
   char section;
   1. Write a single cin statement that will read input into each of these variables.  
      **cin >> age >> pay >> section;**
5. What header files must be included in the following program?  
   int main ()  
   {  
   double amount = 89.7;  
   cout << fixed << showpoint << setprecision (1);  
   cout << setw (8) << amount << endl;  
   return 0;  
   }  
   **#include <iostream>  
   #include <iomapip>**
6. Write a definition statement for a character array named city. It should be large enough to hold a string of 30 characters in length.  
   **char city[32];**
7. Assume the following preprocessor directive appears in a program:  
   #define SIZE 12  
     
   How will the preprocessor rewrite the following lines?
   1. price = SIZE \* unitCost;  
      **price = 12 \* unitCost;**
   2. cout << setw (SIZE0 << 98.7;  
      **error for this line as there is only one paren, SIZE0 is not a variable and neither is 98.7**
   3. cout << SIZE;  
      **cout << 12;**
8. Complete the following table by writing the value of each expression in the value column.
   1. 28 / 4 – 2 **4**
   2. 6 + 12 \* 2 – 8 **22**
   3. 4 + 8 \* 2 **20**
   4. 6 + 17 % 3 – 2 **6**
   5. 2 + 22 \* (9 – 7) **46**
   6. (8 + 7) \* 2 **30**
   7. (16 + 7) % 2 – 1 **0**
   8. 12 / (10 – 6) **3**
   9. (19 – 3) \* (2 + 2) / 4 **16**
9. Write C++ expressions for the following algebraic expressions
   1. A = 12x  
       **a = 12 \* x;**
   2. Z = 5x + 14y + 6k  
      **z = 5 \* x + 14 \* y + 6 \* k;**
   3. Y = x4  
      **y = pow(x, 4);**
   4. G = h + 12/4k  
      **g = h + 12 / 4 \* k;**
   5. G = a3/b2k4  
      **g = pow(a, 3) / pow(b, 2) \* pow(k, 4);**
10. Assume a program has the following variable definitions  
    int units;  
    float mass;  
    double weight;  
      
    and the following statement:  
      
    weight = mass \* units;  
      
    which automatic data type conversions will take place?  
    **units variable will be converted to a float data type. Since weight variable is higher rank than float the weight variable would not be changed as it can store the value from mass and units.**
11. Assume a program has the following variable definitions  
    int a, b = 2;  
    double c = 4.3;  
      
    and the following statement:  
      
    a = b \* c;  
      
    what value will be stored in a?  
    **variable a will have the value 8 stored in it.**
12. Assume that qty and salesReps are both integers. Use a type cast expression to rewrite the following statement so it will no longer perform integer division.  
      
    int qty;  
    int salesReps;  
      
    unitsEach = qty / salesReps;  
    **unitsEach = static\_cast<double> (qty) / salesReps;**
13. Rewrite the following variable definition so the variable is a named constant with the value 12.  
    int rate;  
    **const int rate = 12;**
14. Complete the following table by writing statements with combined assignment operators in the right-hand column. The statements should be equivalent to the statements in the left-hand column.
    1. X = x + 5;  
       **x += 5**
    2. Total = total + subtotal;  
       **total += subtotal;**
    3. Dist = dist / rep;  
       **dist /= rep;**
    4. Ppl = ppl \* period;  
       **ppl = \*= period;**
    5. Inv = inv – shrinkage;  
       **inv -= shrinkage;**
    6. Num = num % 2;  
       **num %= 2**
15. Write a multiple assignment statement that can be used instead of the following group of assignment statements:  
    east = 1;  
    west = 1;  
    north = 1;  
    south = 1;  
    **east = west = north = south = 1;**
16. Replace the following statements with a single statement that initializes sum to 0 at the time it is defined.  
    int sum;  
    sum = 0;  
    **int sum = 0;**
17. Is the following code legal? Why or why not?  
    const int DAYS\_IN\_WEEK;  
    DAYS\_IN\_WEEK = 7;  
    **Honestly not sure, because I wouldn’t write a constant that way, I would write my constants on a single line.**
18. Write a cout statement so the variable divSales is displayed in a field of eight spaces, in fixed-point notation, with a decimal point and two decimal digits.  
    **cout << "divSales: " << setw(8) << fixed << setprecision(2) << divSales << endl;**
19. Write a cout statement so the variable profit is displayed in a field of 12 spaces, in fixed-point notation, with a decimal point and four decimal digits.  
    **cout << " profit: " << setw(12) << fixed << setprecision(2) << profit << endl;**
20. What header file must be included:
    1. To perform mathematical functions like sqrt?  
       **#include <math>**
    2. To use cin and cout?  
       **#include <iostream>**
    3. To use stream manipulators like setprecision?  
       **#include <iomanip>**

Programming challenge 2, 6 & 22 will be in the main.cpp file.