1. Write a single statement to accomplish each of following tasks:
   (a) Use a stream manipulator to ensure that floating-point values print in scientific notation for when using `cout`.
   (b) Use a stream manipulator to set the fill character to `'*'` for printing in field widths larger than the values being output using `cout`.
   (c) Print `6789` right justified in an 8-digit field.

   (a)
   ```
   cout << scientific;
   ```
   (b)
   ```
   cout << setfill( '*' );
   ```
   (c)
   ```
   cout << setw( 8 ) << 6789;
   ```

2. Write a C++ statement that uses the manipulator `setfill` to output a line containing 40 pound signs, i.e., "`########################################`".

   ```
   cout << setfill('#') << setw(40) << "";
   ```

3. Identify error(s), if any, in the following array declarations. If a statement is incorrect, provide the correct statement.
   (a)
   ```
   double weights[100];
   ```
   (b)
   ```
   int age[0..80];
   ```
   (c)
   ```
   int100 list[];
   ```
   (d)
   ```
   double[50] salaries;
   ```

   (a) correct
   (b) should be: `int age[80];`
   (c) should be: `int list[100];`
   (d) should be: `double salaries[50];`

4. Correct the following code so that it correctly sets the value of each element of `myList` to the index of the element.

   ```
   int myList[10];
   for (int i = 1; i <= 10; i--)
       myList[i] = [i];
   ```

   ```
   int myList[10];
   for (int i = 0; i < 10; i++)
       myList[i] = i;
   ```

5. What is stored in the list after the following C++ code executes?

```
int list[10];

list[0] = 2;
list[1] = 3;
for (int i = 2; i < 10; i++)
{
    list[i] = list[i - 1] + list[i - 2];
    if (i > 7)
       list[i] = 2 * list[i] - list[i - 2];
}
```

```
2 3 5 8 13 21 34 55 144 343
```

6. Determine whether the following array declarations are valid. If a declaration is valid, determine the size of the array.

(a) `int list[] = {18, 13, 14, 16};`
(b) `int x[10] = {1,7,5,3,2,8};`
(c) `double y[4] = { 2.0, 5.0, 8.0, 11.0, 14.0} ;`
(d) `int list[7] = {12, 13, , 14, 16, , 8};`

(a) Valid. Array size = 4.
(b) Valid. Array size = 10.
(c) Invalid.
(d) Invalid.

7. Write a single statement for each of the following one-dimensional array operations:

(a) Initialize the 10 elements of integer array `counts` to zero.
(b) Add 1 to each of the 15 elements of the integer array `bonus`.
(c) Read 12 values for `double` array `scores` from the keyboard.

(a)
```
for (int i = 0; i < 10; ++i)
     counts[i] = 0;
```
(b)
```
for (int i = 0; i < 15; ++i)
     bonus[i];
```
(c)
```
for (int i = 0; i < 12; ++i)
     cin >> scores[i];
```

8. Write a code segment that finds the minimum and maximum values contained in a 99-element `double` array `w`.

```
double min, max;
min = max = w[0];
for (int i = 1; i < 99; ++i)
{
    if (w[i] < min)
        min = w[i];
    else if (w[i] > max)
        max = w[i];
}
```

9. Write a code segment that finds the minimum and maximum values contained in a 4-by-6 `int` array `t`. (The declaration for `t` is `int t[4][6];`)

```
double min, max;
min = max = t[0][0];
for (int i = 0; i < 4; ++i)
    for (int j = 0; j < 6; ++j)
    {
        if (t[i][j] < min)
             min = t[i][j];
        else if (t[i][j] > max)
             max = t[i][j];
    }
```

10. Consider the following C++ code:

```
string str1;
string str2;
char ch;
int index;
cin >> str1;
cin >> str2;
cin >> index;
ch = str1[index];
str1[index] = str2[index];
str2[index] = ch;
cout << str1 << " " << str2 << endl;
```

Answer the following questions:

(a) What is the output if the input is `Hello There 2`?
(b) What is the output if the input is `Diamond Gold 0`?
(c) What is the output if the input is `C++ Java 1`?

(a) `Heelo Thlre`
(b) `Giamond Dold`
(c) `Ca+ J+va`

11. What is the value of `str` after the following C++ code is executed?

```
string str1 = "Trip to Hawaii";
string str2 = "Summer or Fall";
string newStr;
newStr = str2 + ' ' + str1;

cout << newStr << endl;
cout << str1 + " in " + str2 << endl;
cout << newStr.length() << endl;
cout << str1.find('H') << endl;
cout << str2.find("or") << endl;
cout << newStr.substr(10, 19) << endl;
cout << newStr.replace(23, 6, "******") << endl;

string str = "C++ Programming";
cout << str << endl;
cout << str.length() << endl;

str[0] = 'J';
str[2] = '$';
```

```
Summer or Fall Trip to Hawaii
Trip to Hawaii in Summer or Fall
29
8
7
Fall Trip to Hawaii
Summer or Fall Trip to ******
C++ Programming
15
J+$ Programming
```

12. Find the error(s) in each of the following, and explain how to correct it (them):

(a) `string string1( 28 );  // construct string1`

(b) `string string2( 'z' );  // construct string2`

Constructors for class string do not exist for integer and character arguments. Other valid constructors should be used—converting the arguments to strings if need be.