Section 10: Arrays

Exercise Solutions:

1) The following Processing statements contain errors. Find and fix them all.

Erroneous code:

```
int[3] intArray;
intArray = {0, 3.14, 6}
intArray[] = new int[7];
intArray[i+1;] = intArray[i];
```

Fixed code:

```
int[] intArray;
intArray = {0, 3, 6};
int[] intArray = new int[7];
intArray[i+1] = intArray[i];
```

2) Write out a Processing statement below to declare and initialize an array that holds the colors of a tricolour flag of your choice (e.g. France, Germany, India, Mexico, Russia). Make sure that you give it an *intuitive* and *legal* array name.

```
color[] mexico = { color(0,255,0), color(255), color(255,0,0) };
```

3) Complete the function below that adds 1 to every index of an array of floats:

```
void addOne( float[] ar ) {
   for( int i = 0; i < ar.length; i = i + 1 ) {
      ar[i] = ar[i] + 1;
   }
}
```

4) Write out Processing code below to declare and initialize a length variable to 50, create an integer array of that length (using the variable), and then use a loop to initialize the array values to their indices (i.e. index 0 holds value 0, index 1 holds value 1, etc.).

```
int len = 50;
int[] ar = new int[len];
for(int i = 0; i < len; i = i + 1) {
   ar[i] = i;
}
```

5) Describe in a sentence what you think the following function accomplishes. Hint: make a simple test array and see what this function does to it! **It *reverses* the ordering of the elements of an array!**

```
void mystery( int[] ar ) {
   int temp, front = 0, back = ar.length - 1;
   while( front < back ) {
      temp = ar[front];
      ar[front] = ar[back];
      ar[back] = temp;
   }
}
```