Chapter 9 Review Questions

1. a. Automatic storage (e.g., `int homer`)
   b. Static with external linkage (e.g., `static secret` outside functions in one file and extern `secret` in the other file)
   c. Static with internal linkage (e.g., `static topsecret`)
   d. Static with no linkage (e.g., `static beencalled`)

2. A using declaration makes specific identifiers within a namespace available, while a using directive makes all identifiers within the namespace available. An example of a using declaration is using `std::cout` while an example of a using declaration is `using namespace std`.

3.
```
#include <iostream>
int main() {
     double x;
     std::cout << "Enter value: ";
     while (! (std::cin >> x) ) {
          std::cout << "Bad input. Please enter a number:
";
          std::cin.clear();
          while (std::cin.get() != '\n')
               continue;
          }
     std::cout << "Value = " << x << std::endl;
     return 0;
     }
```

4.
```
#include <iostream>
int main() {
     using std::cout;
     using std::cin;
     using std::endl;
     double x;
     cout << "Enter value: ";
     while (! (cin >> x) ) {
          cout << "Bad input. Please enter a number: ";
          cin.clear();
          while (cin.get() != '\n')
               continue;
          }
     cout << "Value = " << x << endl;
     return 0;
     }
```

5. I would set up the functions to use static internal linkage as follows:

```
static int average (int num1, int num2); // file 1
static int average (int num1, int num2); // file 2
```

6. The program will display:

```
10
4
0
Other: 10, 1
another(): 10, -4
```

7. The program will display:

```
1
4, 1, 2
2
2
4, 1, 2
2
```