Chapter 4 Review

1.
   ```
   a. char actors[30];
   b. short betsie[100];
   c. float chuck[13];
   d. long double dipsea[64];
   ```

2. Built-in. If created using the array template class, the declarations would be:

   ```
   a. std::array <char, 30> actors;
   b. std::array <short, 100> betsie;
   c. std::array <float, 13> chuck;
   d. std::array <long double, 64> dipsea;
   ```

3. `int posInts[5] = {1, 2, 3, 4, 5};`

4. `int foo = posInts[0] + posInts[4];`

5. `std::cout << ideas[1];`

6. Built-in: `char menuItem[13] = “cheeseburgers”;`

   Template class: `std::array <char, 13> menuItem = “cheeseburgers”;`

7. `string menuItem = “Waldorf Salad”;`

8.
   ```
   struct fish {

         char name[20];
         int weight;
         float length;

   }
   ```

9. `fish foo {“Bass”, 8, 18.5};`

10. `enum Response {no, yes, maybe};`

11.
    ```
    double *foo = &ted;
    std::cout << *foo;
    ```

12.
    ```
    float *foo = treacle;
    std::cout << foo[0] << “, “ << treacle[9];
    ```

13.
    ```
    std::cout << “Enter a positive integer: “;
    std::cin >> posInt;
    int *foo = new int [posInt];
    std::vector<int> bar(posInt);
    ```

14. Yes. It prints the memory address where the string is located.

15.
    ```
    fish *foo = new fish;
    std::cout <<”Enter the name of the fish: “;
    std::cin.get(foo->name);
    ```

16. The program will function as intended unless the year that is input contains more than one word. Ideally, the program should read the year as follows:

    ```
        (std::cin >> year).get();
    ```

17.
    ```
    #include <iostream>
    #include <vector>
    #include <array>

    const int num = 10;

    std::vector <std::string> foo(num);
    std::array <std::string, num> bar;
    ```