Characters

- Built-in data type
- Value: a single character
- Literals: ‘a’, ‘!’, ‘\n’, ‘8’, ...
- Operations:
  - assignment: =
  - compare: ==, <, etc.

```cpp
char ch;
ch = 'a';
if (ch=='A') ... 
```

```cpp
char ch;
cout << "Enter a character: ";
cin >> ch;
```
C-String

• A generic “string” is a sequence of characters
• C-String is a certain way of representing a string in memory
• A C-String is:
  - a sequence of characters stored in consecutive memory locations
  - terminated by a null character (‘\0’)
• A C-String can be stored in a char array.
  - char array is a data type

• string literals are stored in memory as C-Strings:
  - “Jim Kase”, “A00123456”, “$2.35/lb”
• Operations:
  - don’t use = or == on char arrays, won’t work
  - assignment: strcpy(var,value)
  - compare: strcmp(var,value)

```c
cstr = “Biology”; //NOOOO
strcpy(cstr,”Biology”); //YES
cout << “major: “ << cstr; //YES
```

```c
char cstr[10];
cout << “Enter a name: “;
cin >> cstr; // YES
if (cstr==”Math”)... //NOOOO
if (strcmp(cstr,”Math”)==0)... //YES
```
The string class

- string is a data type provided by the C++ library.
  - Specifically it is a class.
- string requires the <string> header file
  - <iostream> may work as well
- To define a string variable:
  - string name1;
  - name1 is called a string object.
- The representation in memory of a string object is hidden from the programmer.

Operations over string objects

- initialization using = with a C-String literal

  ```cpp
  string name1 = "Steve Jobs";
  // can do this with char arrays too
  ```

- assignment using =

  ```cpp
  string name1, name2;
  cout << "Enter a name: ";
  cin >> name1;
  name2 = name1;  // can’t do with char arrays
  ```

- assignment of C-Strings to string objects;

  ```cpp
  string name1;
  name1 = “Andre Johnson”;  // can’t do this with char arrays
  ```
Operations over string objects

- output using `<<`

```cpp
string name1;
name1 = "Steve Jobs";
cout << "Name " << name1 << endl;
```

- input using `>>`

```cpp
string name1;
cout << "Enter your name ";
cin >> name1;
```

- comparing string objects: `<` `<=` `>` `>=` `==` `!=`

  (alphabetical order)

```cpp
string string1, string2;
string1 = "Hello ";
string2 = "World!";
if (string1 < string2)
cout << "Hello comes before World" << endl;
```

- string objects can be compared to C-strings

```cpp
string string1;
cout << "Enter a word: ";
cin >> string1;
if (string1 == "Hello")
cout << "You entered Hello." << endl;
```
Recommended process for Assign7

- Define prototypes and stubs for all four functions
  - if function has a return type, stub should return a dummy value (return true; return 0; etc.)
- Compile (+ test, basically it does nothing)
- Add code for addBook and showList
  - compile + test those two operations
- Add code for remaining two functions:
  - compile + test all operations