

CS150 Intro to CS I

Fall 2017

Character Arrays

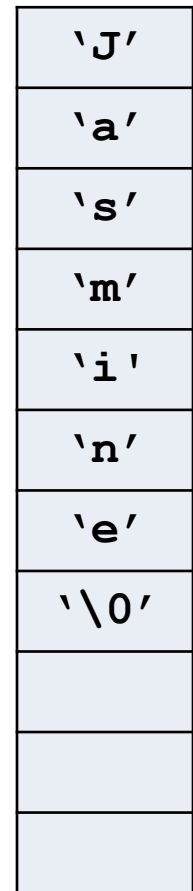
- Reading: pp.554-568

char Arrays

- Character arrays can be used as special arrays called C-strings
- A C-string is just a character array with a null (`\0`) character added at the end
- All string literals are stored as C-strings

Example

```
const int size = 11;  
char name[size] = "Jasmine";
```



Example

```
const int size = 21;  
char name[size];  
  
cout << "enter a name:";  
cin >> name;  
cout << name << endl;
```

Library Functions

- There are many library functions for manipulating and testing C-strings:
 - strlen
 - strcmp
 - strcpy
 - And many others

Example

```
const int size = 21;
char name[size];
int length;

cout << "enter a name:";
cin >> name;
cout << name << endl;

length = strlen(name);

cout << "Length is: " << length << endl;
```

Using `const` as Array Parameters

Page 412

- Arrays are by default passed-by-reference
- Sometimes though, you don't want a function to modify the contents of the array
- Solution:
 - Pass the array as a constant

Example

```
void showValues(const int nums[], int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << nums[i] << endl;
    }
    cout << endl;
}
```

Passing File Objects to Functions

Pages 665-666

- File stream objects must always be passed to functions by reference

```
bool openFile(ifstream &inFile, string name)
```

Practice

- Write a program that will ask the user for a file name and open that file. The program then needs to read a word from the file and then return the number of vowels in that word.
- The program must use the functions on the following slide.

Example

```
bool openFile(ifstream &inFile, string name);  
void getWord(ifstream &inFile, char word[]);  
int countVowels(const char word[]);
```

Working With Characters

- Special functions exist for working with characters
- Remember, `cin >> ch;` reads a single character into `ch` skipping any whitespace
- What is whitespace again?

cin.get ()

- *Sometime we want to read the whitespace!*
- The cin object has a function called get that gets a single character (including whitespace) from the keyboard

```
char ch;  
cout << "Enter a character: ";  
cin.get (ch);  
cout << "Character is: " << ch << endl;
```

cin.get ()

- The following three program segments are *equivalent* for inputting a single character

```
char ch;    // segment #1
cout << "Enter a character: ";
cin.get (ch);
cout << "Character is: " << ch << endl;
```

```
char ch;    // segment #2
cout << "Enter a character: ";
ch = cin.get ();
cout << "Character is: " << ch << endl;
```

```
char ch;    // segment #3
cout << "Enter a character: ";           // NOT REALLY EQUIVALENT
cin >> ch;                               // IGNORES WHITESPACE
cout << "Character is: " << ch << endl;
```

Practice

- What is output for each of the segments below assuming A B C (not ABC) is entered from the keyboard.

```
char ch1, ch2;    // segment #1
cout << "Enter: ";
cin.get (ch1);
cin.get (ch2);
cout << "Characters: " << ch1 << ch2 << endl;
```

```
char ch1, ch2;    // segment #2
cout << "Enter: ";
cin >> ch1 >> ch2;
cout << "Characters: " << ch1 << ch2 << endl;
```


Processing a Line of Text

- We can use `cin.get()` to process an entire line of text including whitespace. What happens if `12345` is entered followed by a return?

```
char ch;
int count = 0;

cout << "Enter a line of text: ";
cin.get (ch);

while ('\n' != ch)
{
    ++count;
    cin.get (ch);
}
cout << "Number of characters is: " << count << endl;
```

cin.peek()

- `char peek();`
 - Look at the next available character but don't remove that character from the stream

```
char nextChar;  
int intValue = -1;  
string textString = "text";
```

```
nextChar = cin.peek();
```

```
if (isdigit(nextChar) )  
{  
    // you read a digit!  
    cin >> intValue;  
}  
else  
{  
    cin >> textString;  
}
```

```
cout << intValue << endl;  
cout << textString << endl;
```

Reading characters from files

- Remember, an ifstream object works similarly to cin;

```
// attempt to read character
readChar = inputFile.get ();

// make sure we read a character
while (!inputFile.eof ())
{
    // process character
    cout << ">" << readChar << "<";

    // attempt to read character again.
    readChar = inputFile.get ();
}
```

Note:
.eof() is only true **after** we tried and failed to read from the file!

Practice

- Remember, an ifstream object works similarly to cin;
- Write a C++ program segment to print the number of lines in the file 'paragraph.txt'