



# LINE I/O

Reading files one line at a time



## LINE INPUT FUNCTIONS: string CLASS

- `istream in("input.txt");`
- `string s;`
- `istream& getline(istream& in, string& s);`
- `istream& getline(istream& in, string& s, char delim);`



## LINE INPUT FUNCTIONS: C-STRING

- `istream in("input.txt");`
- `char s[n];`
- `istream& getline(char* s, int n);`
- `istream& getline(char* s, int n, char delim);`
- `istream& get(char* s, int n);`
- `istream& get(char* s, int n, char delim);`



# SINGLE READ PATTERNS

## string CLASS

```
string line;
while (getline(input, line))
{
    // process line
    output << line << endl;
}
```

## C-STRING

```
char line[100];
while (input.getline(line, 100))
{
    // process line
    output << line << endl;
}
```



## TWO-READ PATTERNS: LINE LENGTH

### string CLASS

```
string line;
getline(input, line);
while (line.length() > 0)
{
    // process line
    cout << line << endl;
    getline(input, line);
}
```

### C-STRING

```
char line[100];
input.getline(line, 100);
while (strlen(line) > 0)
{
    // process line
    cout << line << endl;
    input.getline(line, 100);
}
```



## TWO-READ PATTERNS: eof() FUNCTION

### string CLASS

```
string line;
getline(input, line);
while (!input.eof())
{
    // process line
    cout << line << endl;
    getline(input, line);
}
```

### C-STRING

```
char line[100];
input.getline(line, 100);
while (!input.eof())
{
    // process line
    cout << line << endl;
    input.getline(line, 100);
}
```



## DELIMITED INPUT

- John Smith:123 Elm St.:801-555-1234
- John Smith,123 Elm St.,801-555-1234
- string name;
- getline(in, name, ':');
- char name[20];
- getline(name, 20, ':');
- getline(in, name, ',');
- getline(name, 20, ',');