

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

Character manipulation, newlines, strings

Comp Sci 1570 Introduction to C++



C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

① C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

② Strings

Input

getline()

cin

newlines

Manipulation

C-Strings

Input

cin
 cin.getline()
 cin.get()
 Manipulation

Strings

Input

getline()
 cin
 newlines

Manipulation

① C-Strings

Input

cin
 cin.getline()
 cin.get()

Manipulation

② Strings

Input

getline()
 cin
 newlines

Manipulation

Input to c-string: equivalent effects

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

```
char ntca [50];
```

```
// reads until whitespace
```

```
cin >> ntca; // unsafe
```

```
// reads until '\n', safe
```

```
cin.getline(ntca, sizeof(ntca));
```

```
// reads until '\n', safe
```

```
cin.get(ntca, sizeof(ntca));
```

```
// reads one char, stop is user-defined
```

```
int i = 0;
```

```
do{
```

```
    cin.get(ntca[i]); // or ntca[i]=cin.get();
```

```
    cout << next; i++;
```

```
} while (cin.peek()= '\n'); ntca[i++]='\0';
```

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

```
ntca [50];
```

```
// also can do one character  
cin >> ntca;
```

- cin also leaves a newline character on the stream
- Vulnerable to overflow by walking off the end of the array

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

```
std::cin.getline(ntca, sizeof(ncta), '\n');
```

- Extracts characters from the stream as unformatted input and stores them into s as a c-string, until either the extracted character is the delimiting character, or n characters have been written to s (including the terminating null character).
- The delimiting character is the newline character ('\n') by default, and when found in the input sequence, it is extracted from the input sequence, but discarded and not written to s.
- A null character ('\0') is automatically appended to the written sequence if n is greater than zero, even if an empty string is extracted.
- Optional 3rd parameter – explicit delimiting character: extracting successive characters stops as soon as the next character to extract compares equal to this.

C-Strings

Input
 cin
 cin.getline()
cin.get()
 Manipulation

Strings

Input
 getline()
 cin
 newlines
 Manipulation

```
// similar to getline ,  

// but leaves '\n' on stream  

std :: cin . get ( ntca , sizeof ( ncta ) , '\n' );
```

- Extracts characters from the stream and stores them in s as a c-string, until either (n-1) characters have been extracted or the delimiting character is encountered: the delimiting character being either the newline character ('\n') or delim (if this argument is specified).
- The delimiting character is not extracted from the input sequence if found, and remains there as the next character to be extracted from the stream (see getline for an alternative that does discard the delimiting character).
- A null character ('\0') is automatically appended to the written sequence if n is greater than zero, even if an empty string is extracted.

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

```
char c;
```

```
// also can do one character  
std::cin.get(c)
```

```
// or this  
c = std::cin.get();
```

- Extracts a single character from the stream.
- The character is either returned (first signature), or set as the value of its argument (second signature).

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

① C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

② Strings

Input

getline()

cin

newlines

Manipulation

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

The `putback()` function will allow you to put a character back into the input stream:

```
cin . putback ( char_var );
```

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

The `peek()` function will allow you to know what the next character in the input stream is without extracting from that stream.

C-Strings

Input

cin
 cin.getline()
 cin.get()

Manipulation

Strings

Input

getline()
 cin
 newlines

Manipulation

- `toupper(char)` returns the uppercase of arg sent
`toupper('a');` – > 'A'
- `tolower(char)` similar
- `isupper(char)` returns bool: true if uppercase `isupper('a');`
 – > false
- `islower(char)` similar
- `isalpha(char)` similar
- `isdigit(char)` similar
- `ispunct(char)` returns bool: true if punctuation
`ispunct('!');` – > true
- `isspace(char)` returns bool: true if whitespace space,
 newline, tab

Check out examples

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

① C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

② Strings

Input

getline()

cin

newlines

Manipulation

C-Strings

Input

cin
 cin.getline()
 cin.get()

Manipulation

Strings

Input

getline()
 cin
 newlines

Manipulation

① C-Strings

Input

cin
 cin.getline()
 cin.get()

Manipulation

② Strings

Input

getline()
 cin
 newlines

Manipulation

String input: equivalent examples

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

```
string mystring;
```

```
// reads until whitespace, and
```

```
// leaves '\n' on stream
```

```
cin >> mystring;
```

```
// reads until '\n', discards it
```

```
getline(std::cin, mystring);
```

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

```
getline(std::cin, mystring);
```

```
getline(std::cin, mystring, '\\n');
```

- Extracts characters from input stream and stores them into mystring until the delimitation character delim is found (or the newline character, '\\n')
- The extraction also stops if the end of file is reached in is or if some other error occurs during the input operation.
- If the delimiter is found, it is extracted and discarded (i.e. it is not stored and the next input operation will begin after it).
- Note that any content in str before the call is replaced by the newly extracted sequence.
- Memory safe

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

`cin >> mystring`

- cin also leaves a newline character on the stream
- NOT vulnerable to overflow by walking off the end of the array

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

Check out code examples

```
std::cin.ignore(32767, '\n');
```

If you used one of the methods which leaves newlines on the stream, and you are using a different method afterwards that reads only until a newline, then you need this!

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

① C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

② Strings

Input

getline()

cin

newlines

Manipulation

C-Strings

Input

cin

cin.getline()

cin.get()

Manipulation

Strings

Input

getline()

cin

newlines

Manipulation

- Check out the code!