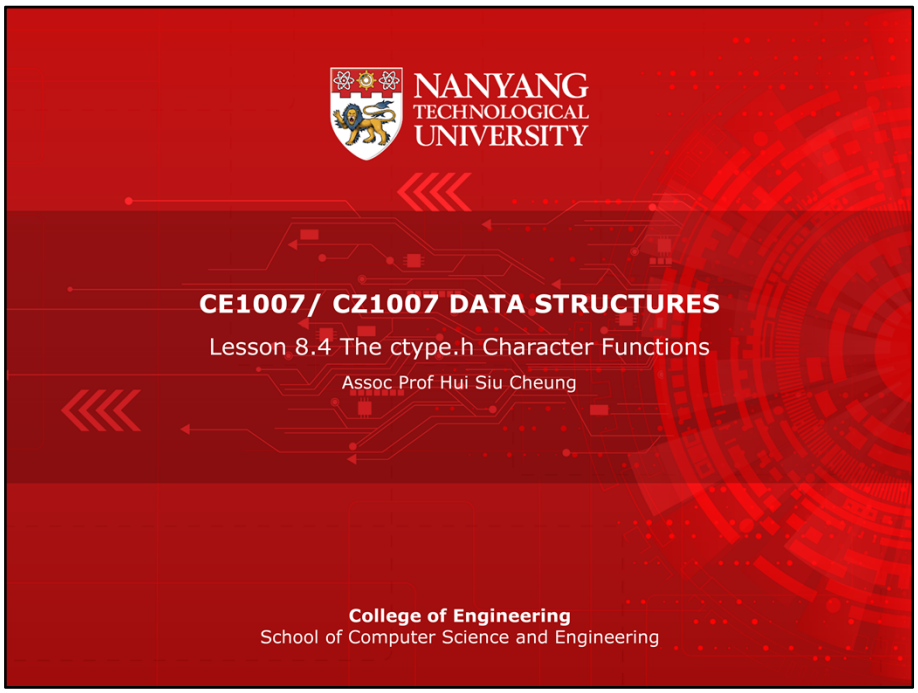
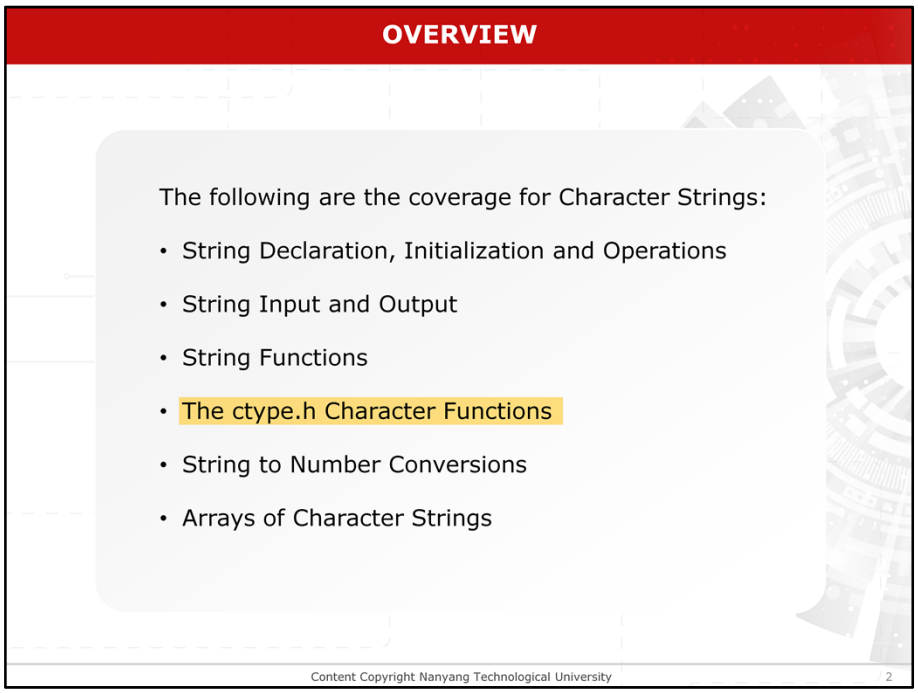
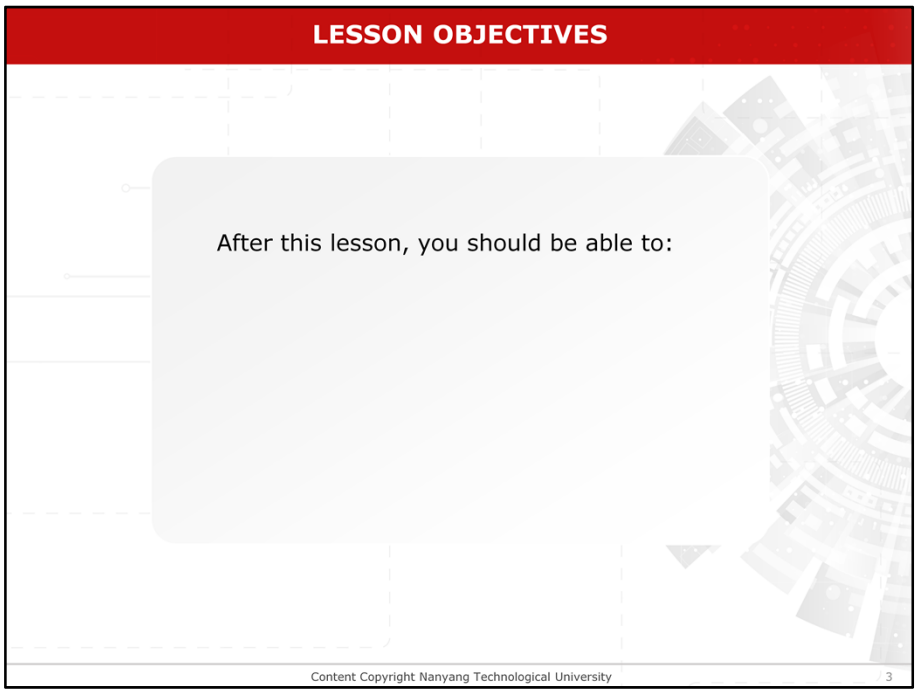
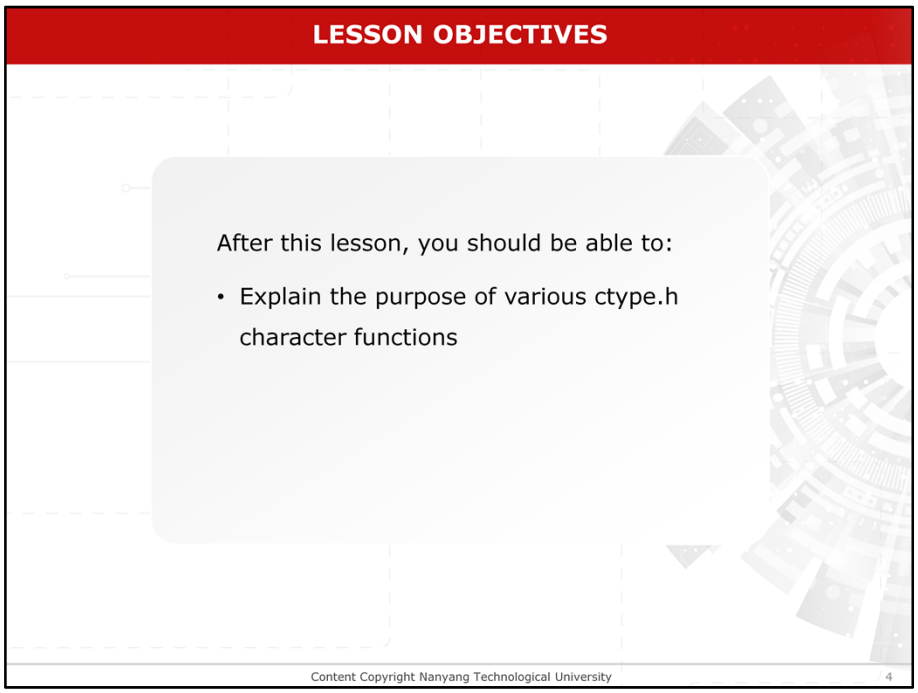

NANYANG
TECHNOLOGICAL
UNIVERSITY
CE1007/ CZ1007 DATA STRUCTURES
Lesson 8.4 The ctype.h Character Functions
Assoc Prof Hui Siu Cheung
College of Engineering
School of Computer Science and Engineering



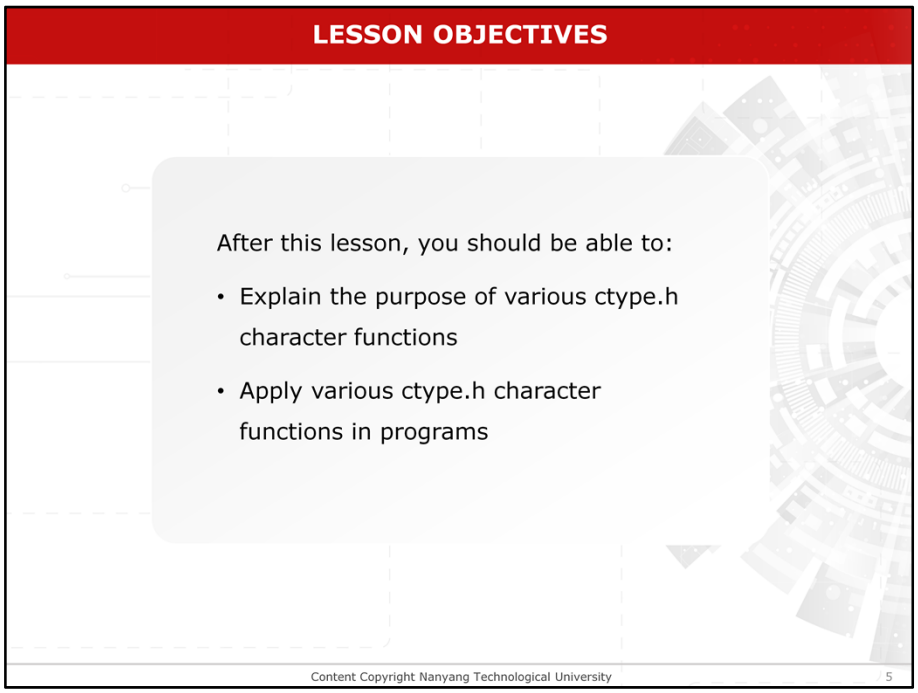

There are 6 main sections to cover for Character Strings as shown. This video lesson focuses on the fourth part on the Character Functions



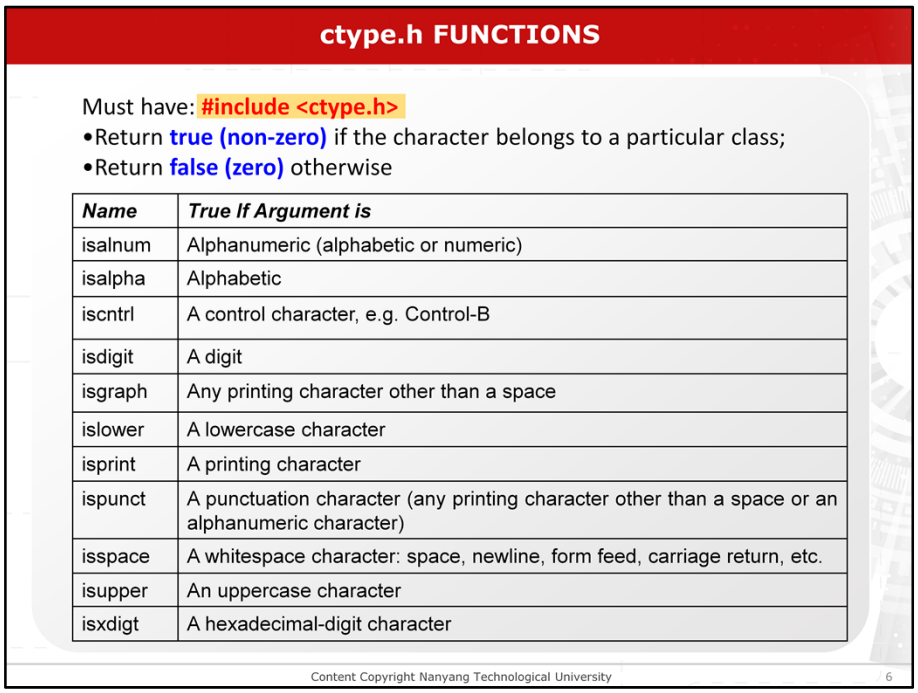

After this lesson, you should be able to:




Explain the purpose of various character functions




Apply various character functions in programs




C also contains the character processing library, whose functions are declared in the header file. These functions are used to test the nature of a character. It returns *true* if the condition being tested is satisfied, or *false* otherwise. To use these functions, we must include the header file in the programs. Some of the most commonly used functions are given in the table.

## ctype.h FUNCTIONS

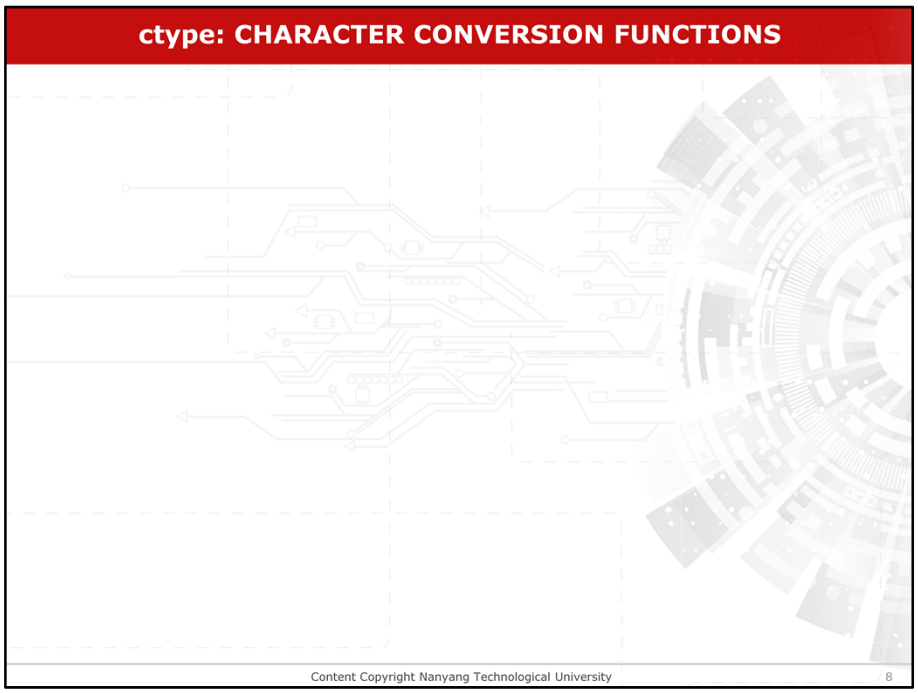
Must have: **#include <ctype.h>**

- Return **true (non-zero)** if the character belongs to a particular class;
- Return **false (zero)** otherwise

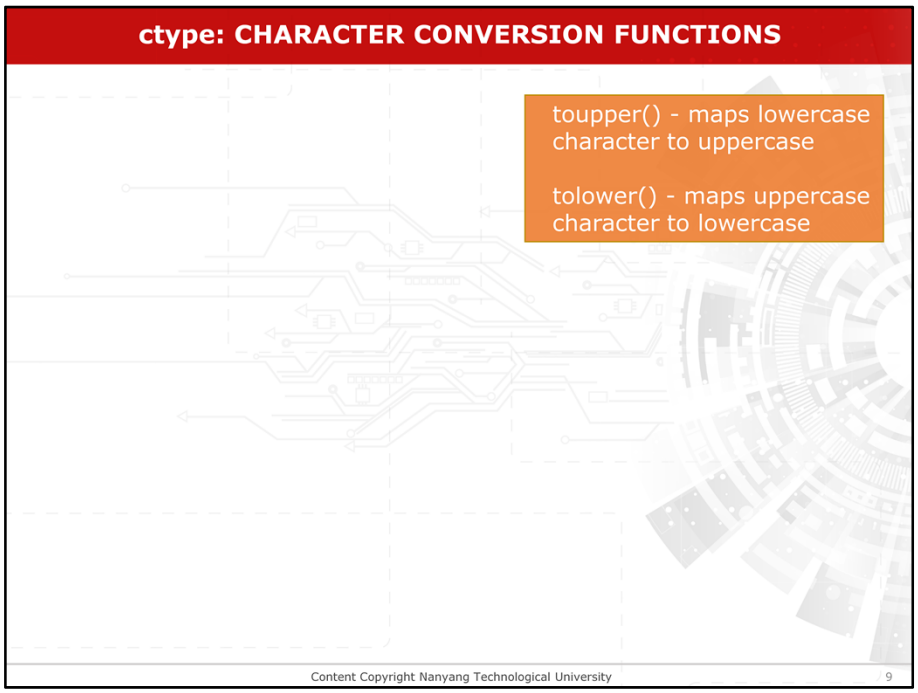
| *Name* | *True If Argument is* |
|---|---|
| isalnum | Alphanumeric (alphabetic or numeric) |
| isalpha | Alphabetic |
| iscntrl | A control character, e.g. Control-B |
| isdigit | A digit |
| isgraph | Any printing character other than a space |
| islower | A lowercase character |
| isprint | A printing character |
| ispunct | A punctuation character (any printing character other than a space or an alphanumeric character) |
| isspace | A whitespace character: space, newline, form feed, carriage return, etc. |
| isupper | An uppercase character |
| isxdigt | A hexadecimal-digit character |



The character testing functions are very useful. For example, when an input might contain any sequence of input characters, we can use the function such as **is lower()**, **is upper()**, **is digit()**, **is alpha()**, **is alpha numeric()** or **is space()** to test each input character and then process the character accordingly.

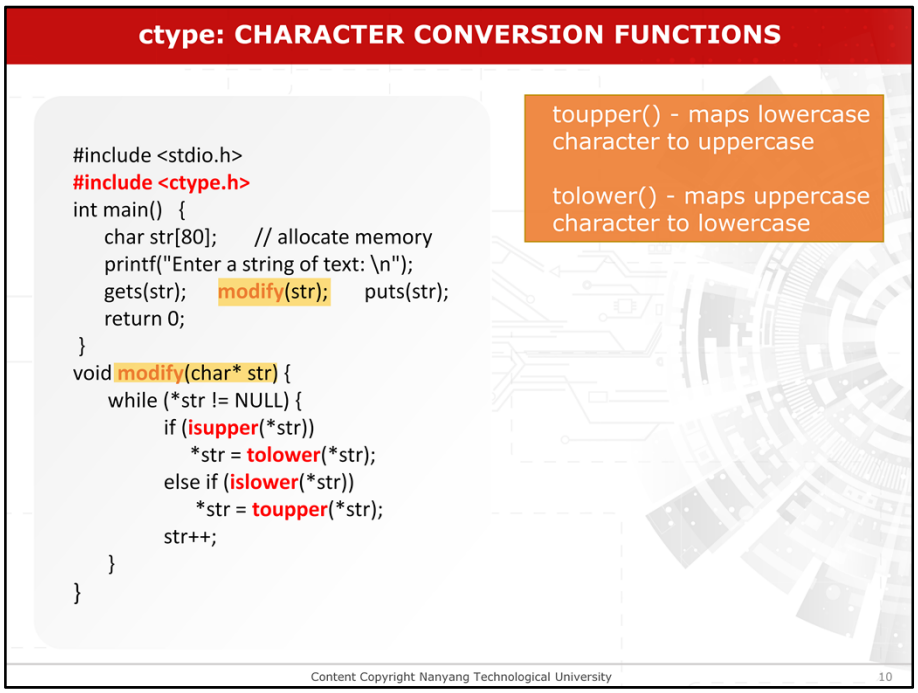



The Functions: Character Conversion Functions




In addition to the functions that test characters in , there are several character conversion functions for converting characters. The two most commonly used are **to upper()** and **to lower()**.

The function **to upper()** converts lowercase characters to uppercase, while the function **to lower()** converts uppercase characters to lowercase. These two functions are commonly used to test character input, and convert all of them into either lowercase or uppercase, so that the program is not sensitive to the case of the letters the user enters.




In the program, the function **modify()** aims to convert lower case letters to upper case letters and vice versa from an input string.

## ctype: CHARACTER CONVERSION FUNCTIONS

```
#include <stdio.h>
#include <ctype.h>
int main()  {
    char str[80];     // allocate memory
    printf("Enter a string of text: \n");
    gets(str);     modify(str);     puts(str);
    return 0;
 }
void modify(char* str) {
     while (*str != NULL) {
            if (isupper(*str))
                *str = tolower(*str);
            else if (islower(*str))
                 *str = toupper(*str);
            str++;
     }
}
```

toupper() - maps lowercase character to uppercase

tolower() - maps uppercase character to lowercase

## ctype: CHARACTER CONVERSION FUNCTIONS

```
#include <stdio.h>
#include <ctype.h>
int main()  {
    char str[80];      // allocate memory
    printf("Enter a string of text: \n");
    gets(str);     modify(str);     puts(str);
    return 0;
 }
void modify(char* str) {
     while (*str != NULL) {
            if (isupper(*str))
                *str = tolower(*str);
            else if (islower(*str))
                 *str = toupper(*str);
            str++;
     }
}
```

toupper() - maps lowercase character to uppercase

tolower() - maps uppercase character to lowercase


**User Input**

Enter a string of text:

## ctype: CHARACTER CONVERSION FUNCTIONS

```
#include <stdio.h>
#include <ctype.h>
int main()  {
    char str[80];      // allocate memory
    printf("Enter a string of text: \n");
    gets(str);    modify(str);      puts(str);
    return 0;
 }
void modify(char* str) {
     while (*str != NULL) {
            if (isupper(*str))
                *str = tolower(*str);
            else if (islower(*str))
                 *str = toupper(*str);
            str++;
     }
}
```

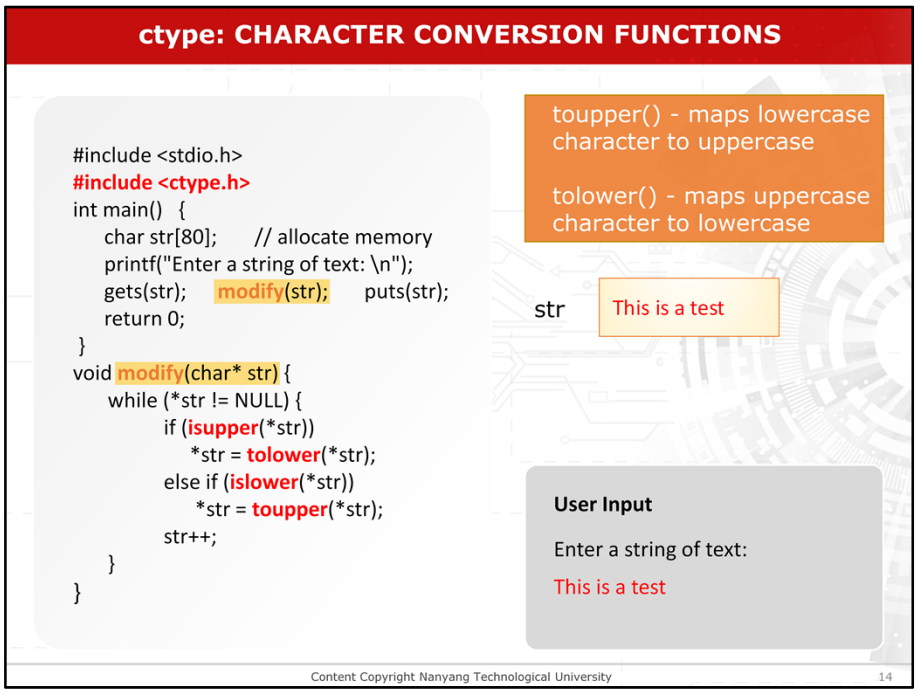
toupper() - maps lowercase character to uppercase

tolower() - maps uppercase character to lowercase

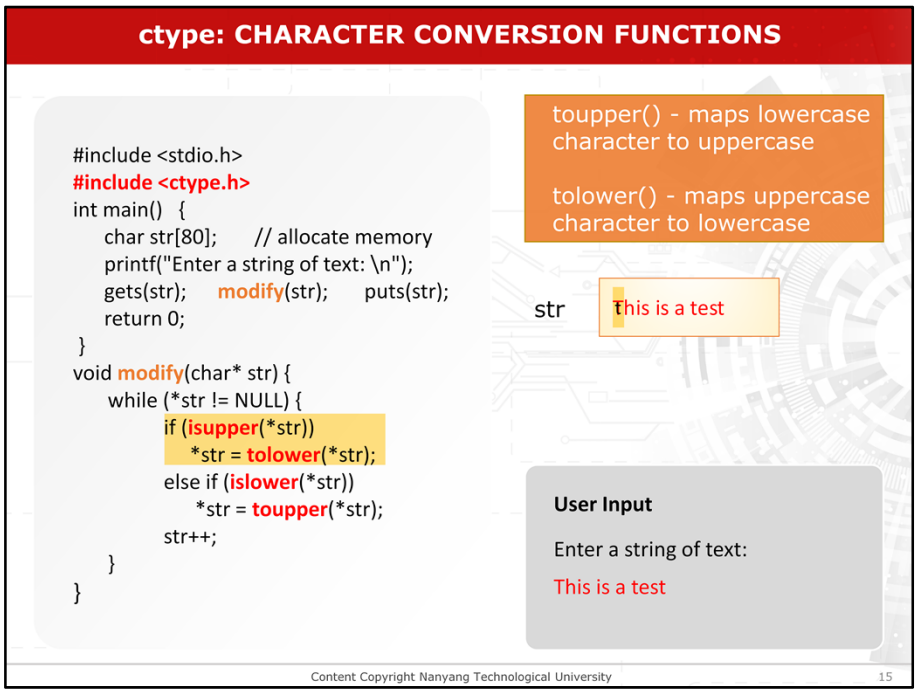

**User Input**

Enter a string of text:

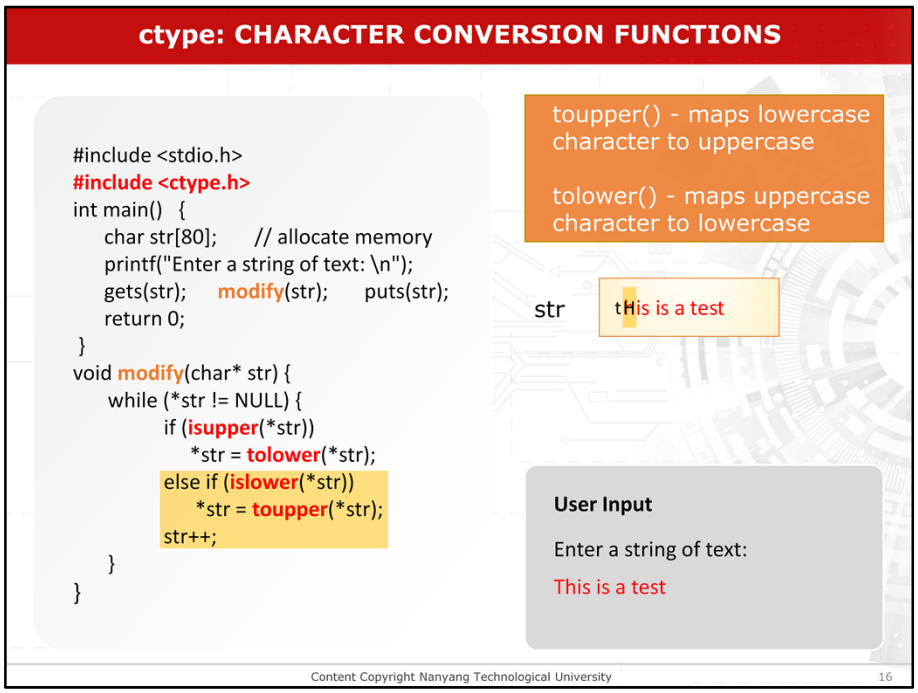
This is a test



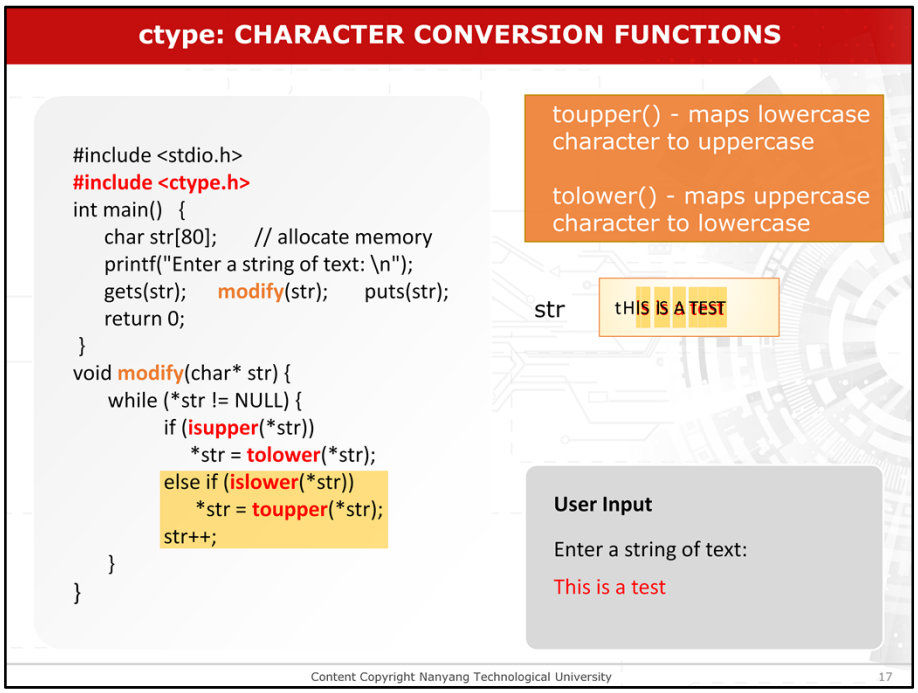

In the implementation of the **modify()** function, it uses the functions **is upper()**, **is lower()**, **to lower()** and **to upper()** for implementing character conversion.




If a character in the input string is tested to be in uppercase, it will be converted into lowercase using the function **to lower()**.




Similarly, if a character in the input string is tested to be in lowercase, it will be converted to uppercase using the function **to upper()**.


ctype: CHARACTER CONVERSION FUNCTIONS
#include <stdio.h>
#include <ctype.h>
int main() {
char str[80]; // allocate memory
printf("Enter a string of text: \n");
gets(str); modify(str); puts(str);
return 0;
}
void modify(char* str) {
while (*str != NULL) {
if (isupper(*str))
*str = tolower(*str);
else if (islower(*str))
*str = toupper(*str);
str++;
}
}
toupper() - maps lowercase character to uppercase
tolower() - maps uppercase character to lowercase
str
tHIS IS A TEST
User Input
Enter a string of text:
This is a test
Content Copyright Nanyang Technological University
17

[no audio]

## ctype: CHARACTER CONVERSION FUNCTIONS

```
#include <stdio.h>
#include <ctype.h>
int main()  {
    char str[80];      // allocate memory
    printf("Enter a string of text: \n");
    gets(str);     modify(str);     puts(str);
    return 0;
 }
void modify(char* str) {
     while (*str != NULL) {
            if (isupper(*str))
                *str = tolower(*str);
            else if (islower(*str))
                 *str = toupper(*str);
            str++;
     }
}
```

toupper() - maps lowercase character to uppercase

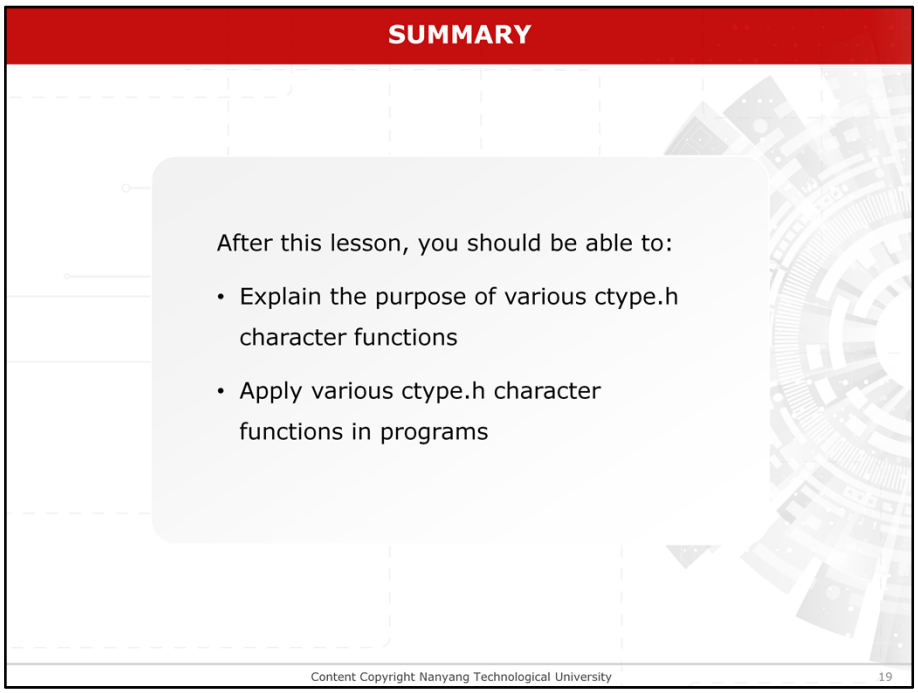
tolower() - maps uppercase character to lowercase

str | tHIS IS A TEST

**User Input**

Enter a string of text:

This is a test




In summary, after viewing this video lesson, you should be able to do the points listed.