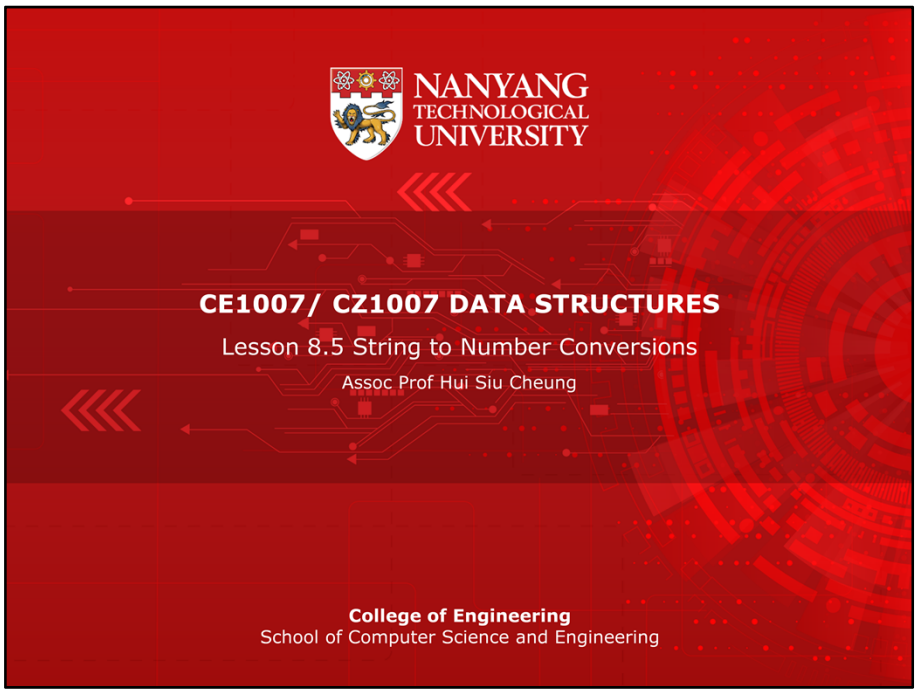
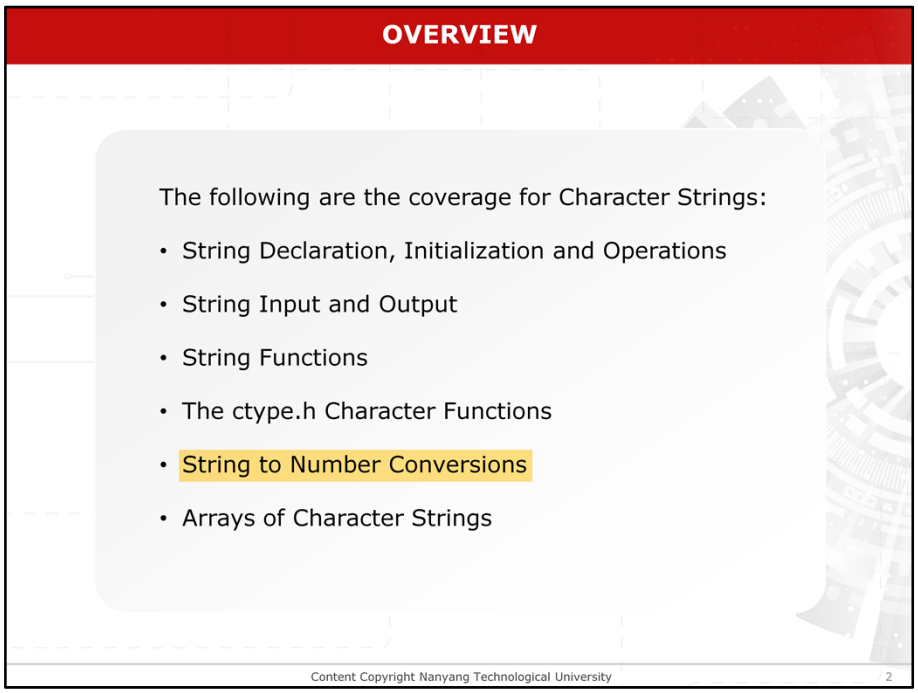
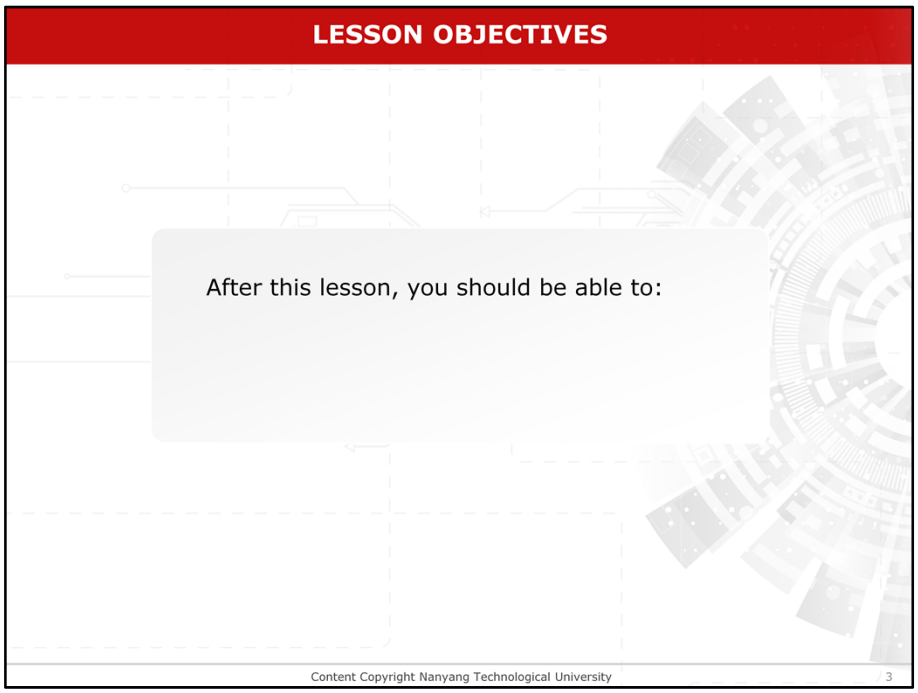
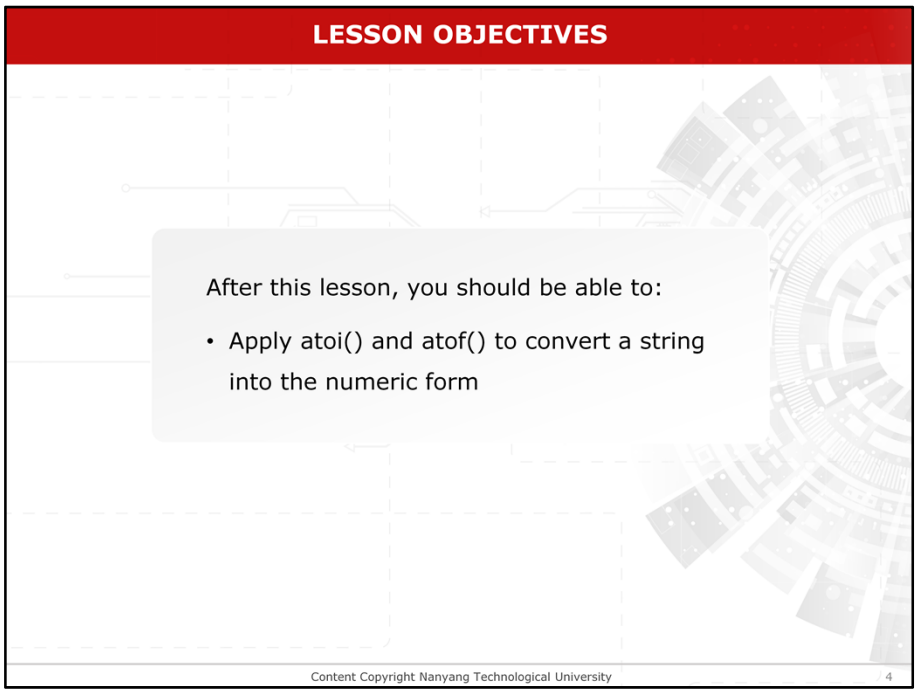

NANYANG
TECHNOLOGICAL
UNIVERSITY
CE1007/ CZ1007 DATA STRUCTURES
Lesson 8.5 String to Number Conversions
Assoc Prof Hui Siu Cheung
College of Engineering
School of Computer Science and Engineering



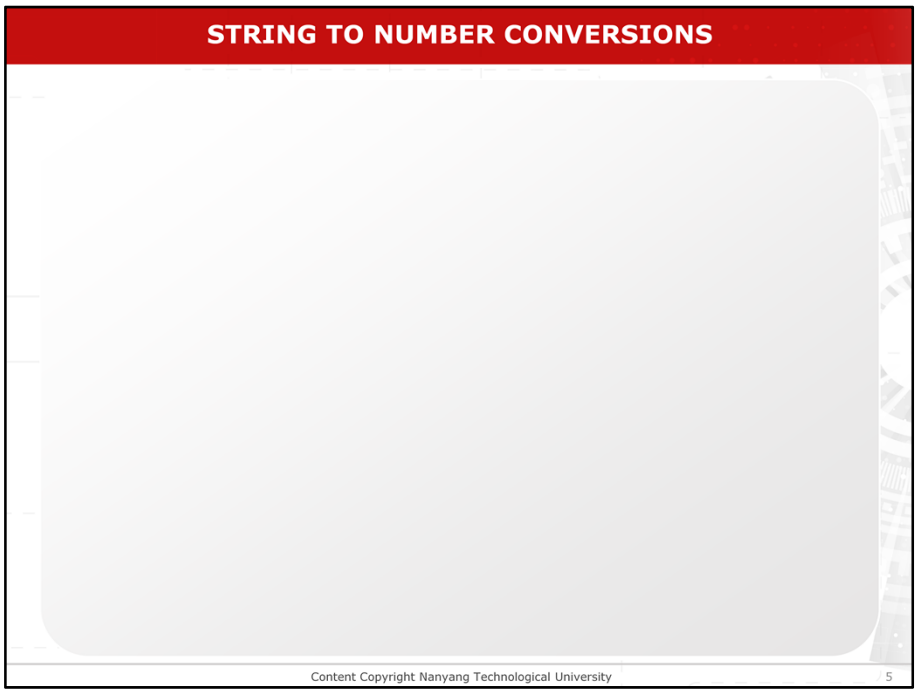
There are 6 main sections to cover for Character Strings as shown. This video lesson focuses on the fifth part on



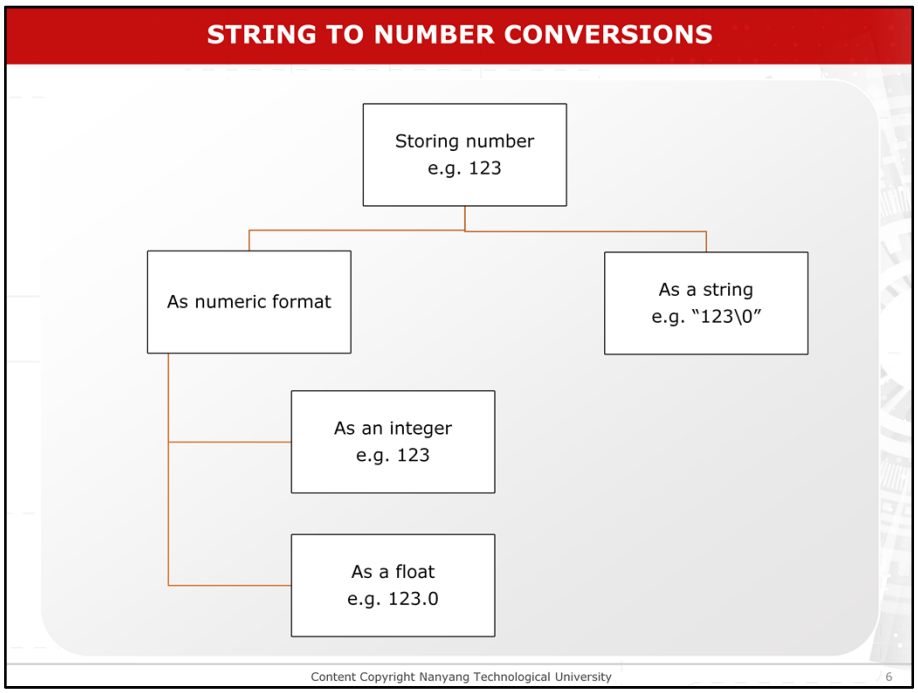
After this lesson, you should be able to:



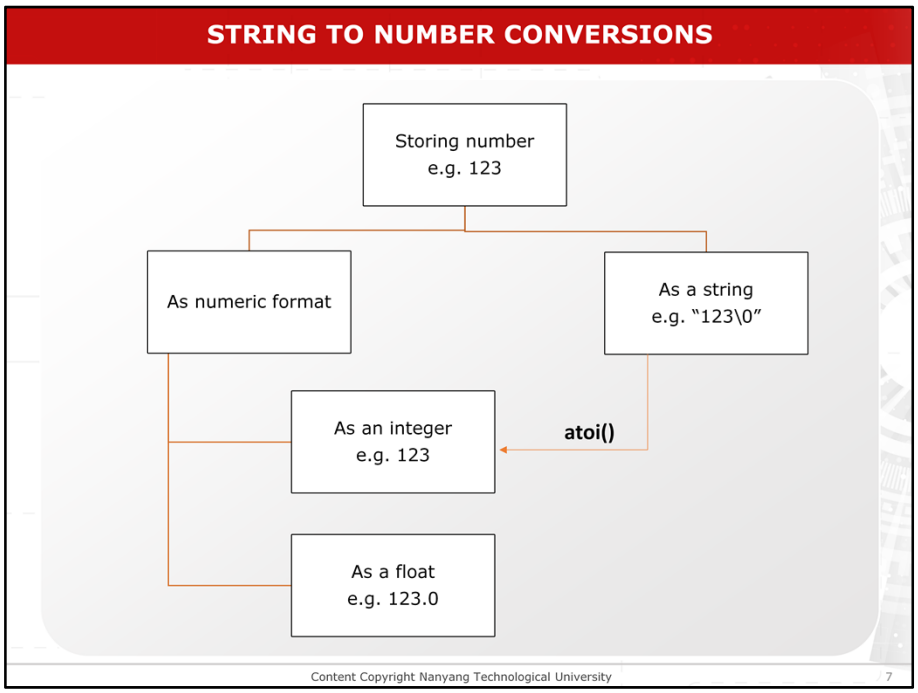
Apply “a” to “i” and “a” to “f” to convert a string into the numeric form



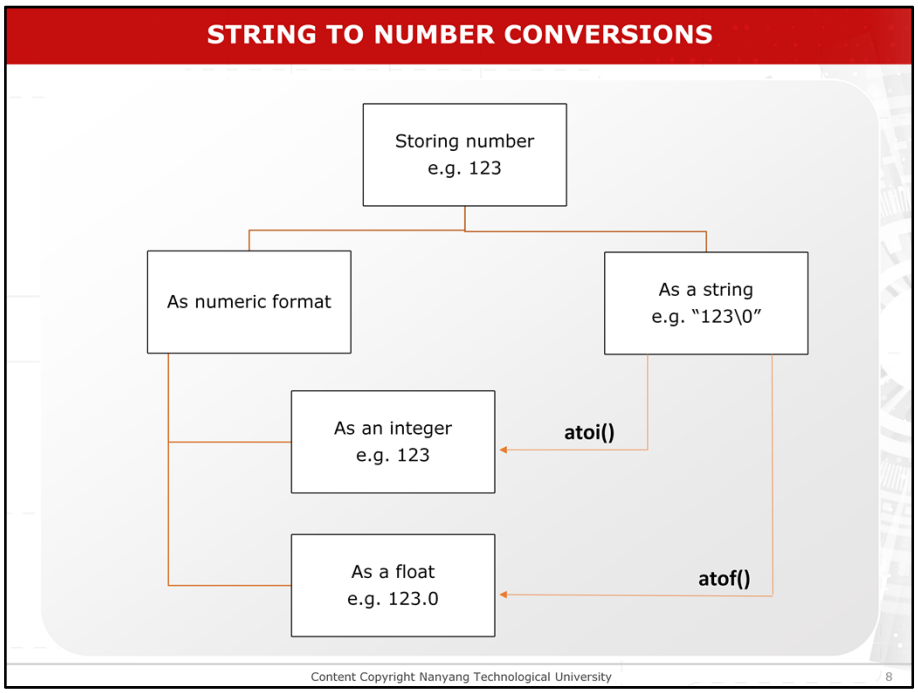
String to Number Conversions



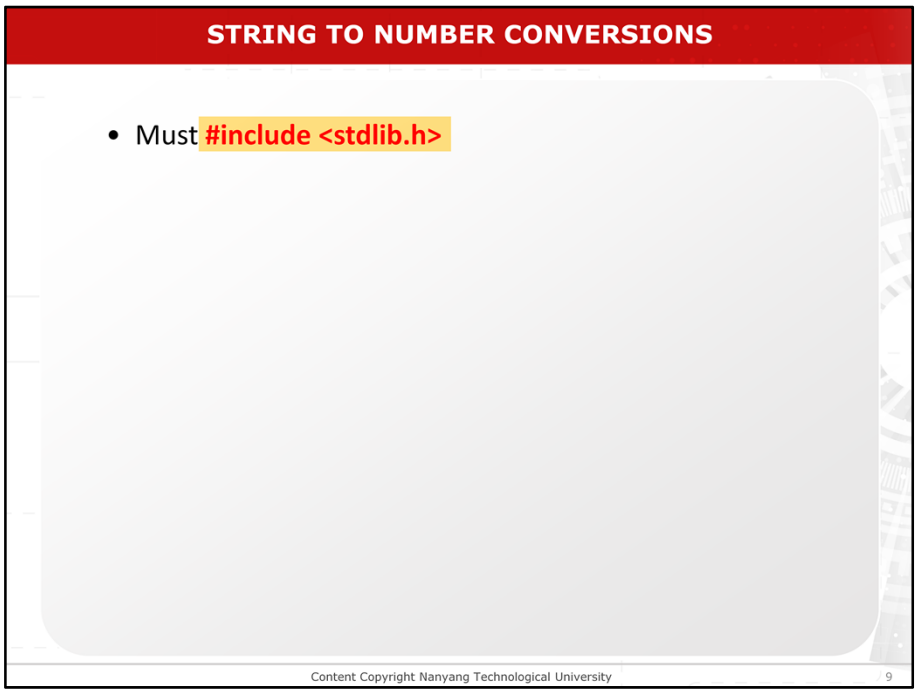
There are two ways to store a number. It can be stored in numeric form or as strings. For example, the number 123 can be stored as a string consisting of '1', '2', '3' and the terminating null character. Sometimes, it is convenient to read in the numerical data as a string and convert it into the numeric form.



To do this, C provides the functions: “a” to “i” and “a” to “f”
The “a” to “i” function converts a character string into an integer and returns the integer value.

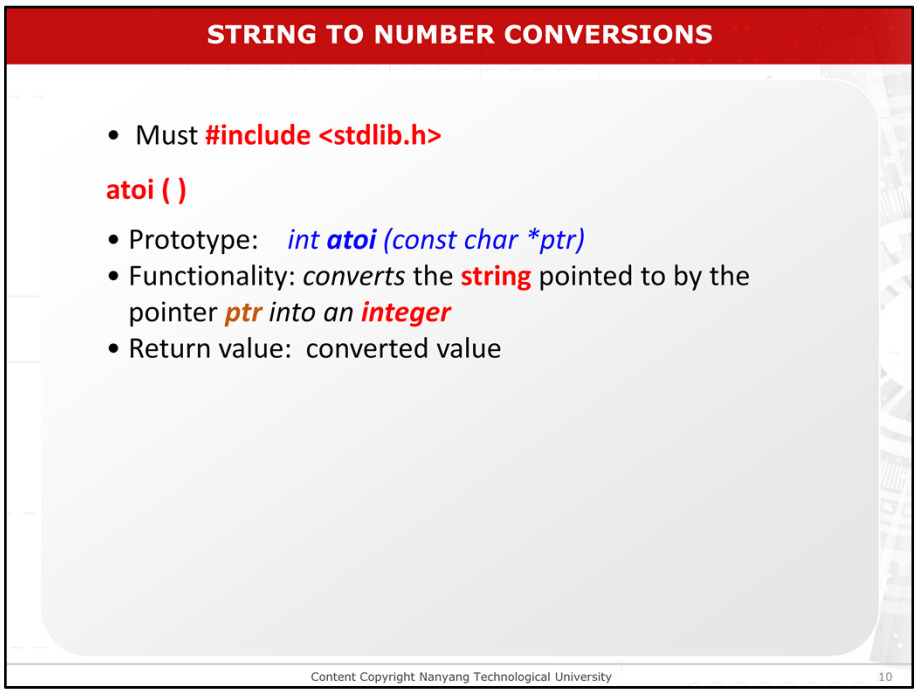


The “a” to “f” function converts a string into a double precision floating point value.



To use these functions, we must include the *standard library dot h* file in the program:

pound include standard library dot h()

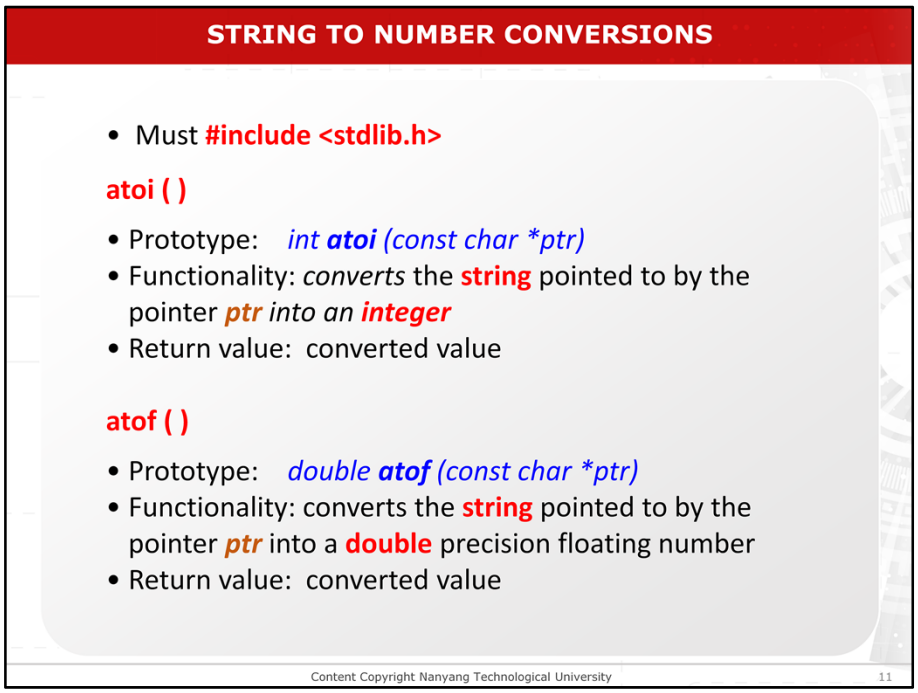


The function prototype is

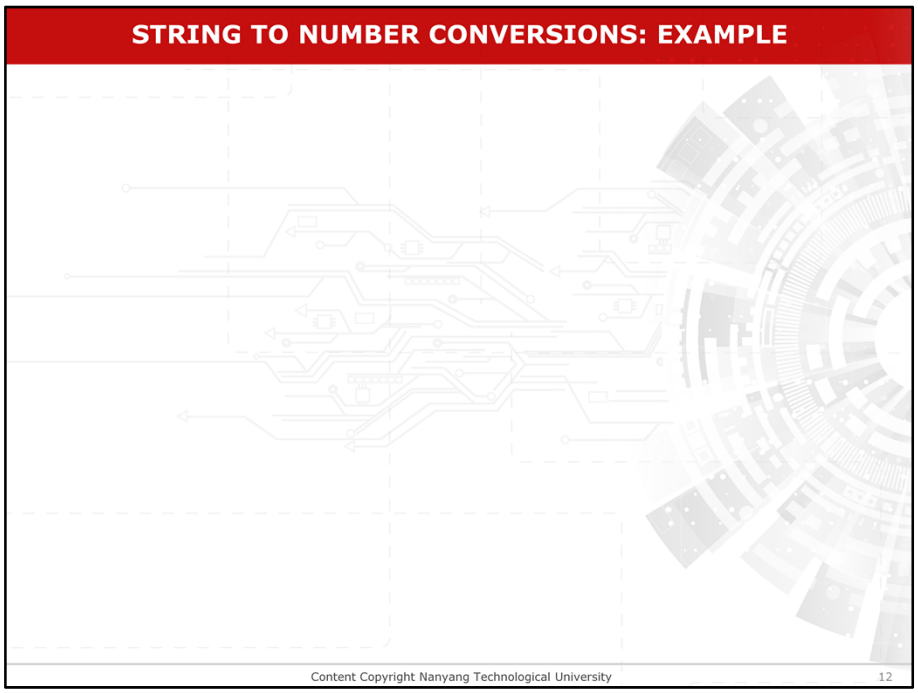
integer "a" to "i" **(constant character asterisk pointer);**

The **"a" to "i"** function converts a character string into an integer and returns the integer value.

The **"a" to "i"**function processes the digits in the string and stops when the first non digit character is encountered. Leading blanks are ignored and leading algebraic sign + or - can be recognized.



The **"a" to "f"** function converts a string into a double precision floating point value.

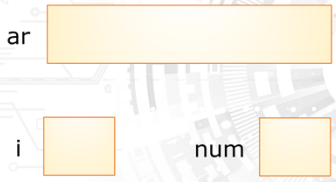


This example uses the **“a” to “i”** function for string to integer number conversion.

STRING TO NUMBER CONVERSIONS: EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
  char  ar[80];
  int  i,   num;

  scanf("%s", ar);          //  read input string
  i=0;
  while (isdigit(ar[i])    // check digit in string
     i++;
  if (ar[i] != '\0')          // if not a null character
     printf("The input is not a number\n");
     /* for example, “1a2” */
  else {
     num = atoi(ar);
     printf("Input is %d\n", num);
  }
```

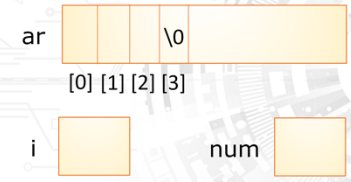


Variables are declared.

STRING TO NUMBER CONVERSIONS: EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
  char  ar[80];
  int  i,   num;

  scanf("%s", ar);          //  read input string
  i=0;
  while (isdigit(ar[i])    // check digit in string
     i++;
  if (ar[i] != '\0')          // if not a null character
     printf("The input is not a number\n");
     /* for example, “1a2” */
  else {
     num = atoi(ar);
     printf("Input is %d\n", num);
  }
```



User Input

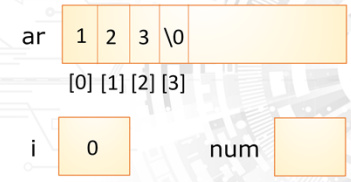
1 2 3

Array ar stores the input string 123

STRING TO NUMBER CONVERSIONS: EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
  char  ar[80];
  int  i,   num;

  scanf("%s", ar);          //  read input string
  i=0;
  while (isdigit(ar[i]))    // check digit in string
     i++;
  if (ar[i] != '\0')          // if not a null character
     printf("The input is not a number\n");
     /* for example, “1a2” */
  else {
     num = atoi(ar);
     printf("Input is %d\n", num);
  }
```



User Input

1 2 3

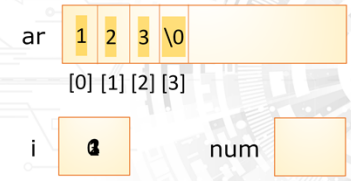
Intialise i to 0

STRING TO NUMBER CONVERSIONS: EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
  char  ar[80];
  int  i,   num;

  scanf("%s", ar);          //  read input string
  i=0;
  while (isdigit(ar[i]))    // check digit in string
     i++;
  if (ar[i] != '\0')        // if not a null character
     printf("The input is not a number\n");
     /* for example, “1a2” */
  else {
     num = atoi(ar);
     printf("Input is %d\n", num);
  }
```

isdigit() returns true if argument is a digit



User Input

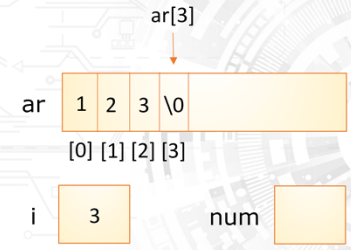
1 2 3

Using is digit function to check whether its digit in string

STRING TO NUMBER CONVERSIONS: EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
  char  ar[80];
  int  i,   num;

  scanf("%s", ar);          //  read input string
  i=0;
  while (isdigit(ar[i]))    // check digit in string
     i++;
  if (ar[i] != '\0')           // if not a null character
     printf("The input is not a number\n");
     /* for example, “1a2” */
  else {
     num = atoi(ar);
     printf("Input is %d\n", num);
  }
```



User Input

1 2 3

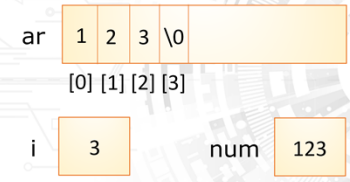
Need to check whether the entire string is made of all digits

STRING TO NUMBER CONVERSIONS: EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
  char  ar[80];
  int  i,   num;

  scanf("%s", ar);          //  read input string
  i=0;
  while (isdigit(ar[i]))    // check digit in string
     i++;
  if (ar[i] != '\0')          // if not a null character
     printf("The input is not a number\n");
     /* for example, "1a2" */
  else {
     num = atoi(ar);
     printf("Input is %d\n", num);
  }
```

atoi() converts string pointed by pointer into an integer



i 3 num 123

User Input

1 2 3

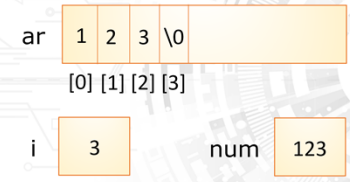
"a" to "i" function converts string pointed by pointer into an integer

STRING TO NUMBER CONVERSIONS: EXAMPLE

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main()
{
  char  ar[80];
  int  i,   num;

  scanf("%s", ar);          //  read input string
  i=0;
  while (isdigit(ar[i]))    // check digit in string
     i++;
  if (ar[i] != '\0')          // if not a null character
     printf("The input is not a number\n");
     /* for example, “1a2” */
  else {
     num = atoi(ar);
     printf("Input is %d\n", num);
  }
```

atoi() converts string pointed by pointer into an integer



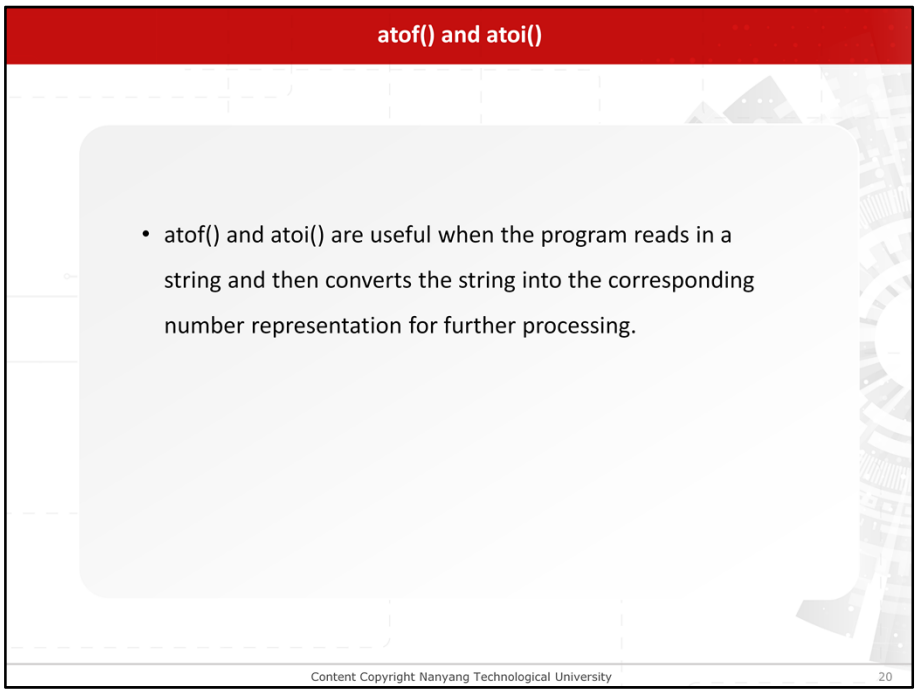
i 3 num 123

User Input

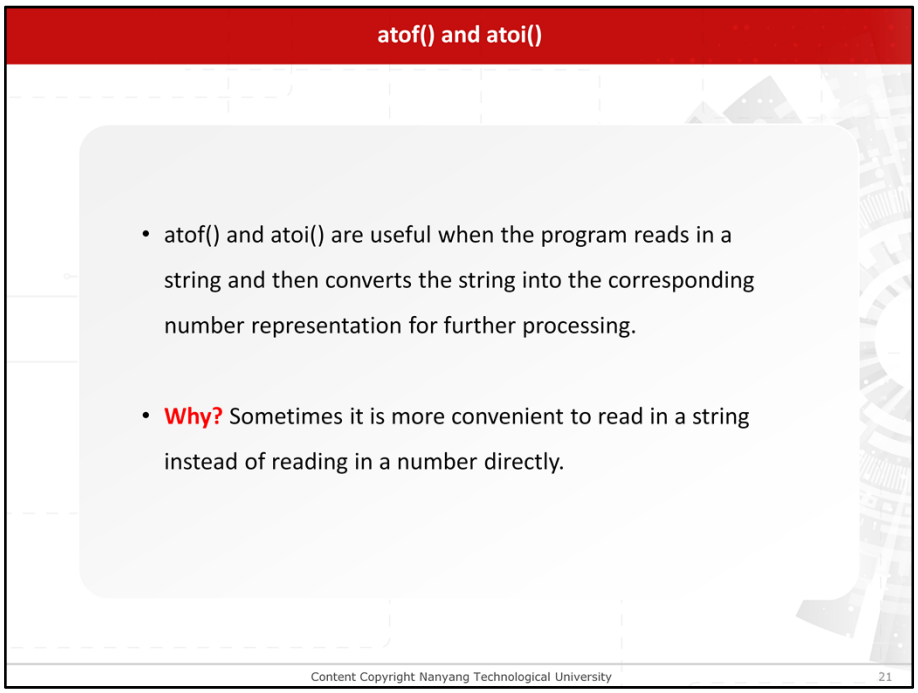
1 2 3

Input is 123

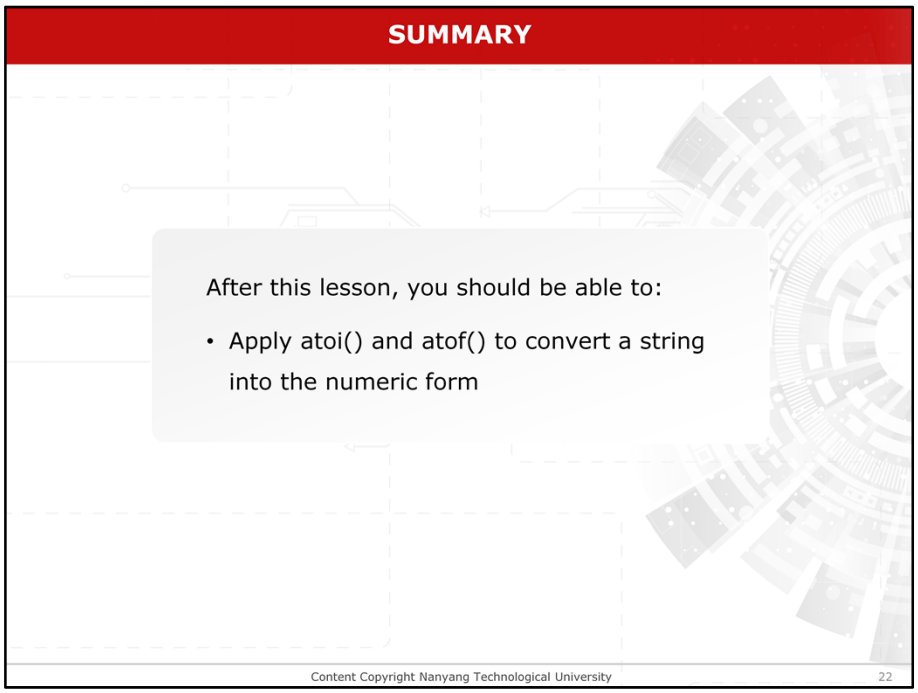
Num is an integer



“a” to “f” and **“a” to “i”** are useful when the program reads in a string and then converts the string into the corresponding number representation for further processing.



Because sometimes it is more convenient to read in a string instead of reading in a number directly.



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