

Modules Testing

Version #2



November 30, 2024

GROUP 33

Michaela & Vadeeha

Table of Contents

[Main 2](#_Toc183615924)

[Fundamentals 5](#_Toc183615925)

[Manipulating 6](#_Toc183615926)

[Converting 7](#_Toc183615927)

[Tokenizing 8](#_Toc183615928)

# Main

1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
1  
\*\*\* Start of Indexing Strings Demo \*\*\*  
Type not empty string (q - to quit):  
Group 33 !!  
Type the character position within the string:  
10  
The character found at 10 position is '!'  
Type not empty string (q - to quit):  
q  
\*\*\* End of Indexing Strings Demo \*\*\*  
1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
2  
\*\*\* Start of Concatenating Strings Demo \*\*\*  
Type the 1st string (q - to quit):  
Version  
Type the 2nd string:  
One  
Concatenated string is 'VersionOne'  
Type the 1st string (q - to quit):  
q  
\*\*\* End of Concatenating Strings Demo \*\*\*

1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
3  
\*\*\* Start of Converting Strings to int Demo \*\*\*  
Type an int numeric string (q - to quit):  
101  
Converted number is 101  
Type an int numeric string (q - to quit):  
q  
\*\*\* End of Converting Strings to int Demo \*\*\*

1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
4  
\*\*\* Start of Tokenizing Words Demo \*\*\*  
Type a few words separated by space (q - to quit):  
Seneca College  
Word #1 is 'Seneca'  
Word #2 is 'College'  
Type a few words separated by space (q - to quit):  
q  
\*\*\* End of Tokenizing Words Demo \*\*\*

1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
a  
Invalid option, please try again.  
1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
!  
Invalid option, please try again.  
1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
8  
Invalid option, please try again.  
1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
-6  
Invalid option, please try again.  
1 - Fundamentals  
2 - Manipulating  
3 - Converting  
4 - Tokenizing  
0 - Exit  
Which module to run?  
0  
Exiting the program.

# Fundamentals

\*\*\* Start of Indexing Strings Demo \*\*\*

Type not empty string (q - to quit):

q

\*\*\* End of Indexing Strings Demo \*\*\*

\*\*\* Start of Measuring String Demo \*\*\*

Type a string (q - to quit):

Group 33 !!

The length of 'Group 33 !!' is 11 characters

Type a string (q - to quit):

Michaela&Vadeeha

The length of 'Michaela&Vadeeha' is 16 characters

Type a string (q - to quit):

12345

The length of '12345' is 5 characters

Type a string (q - to quit):

VersionTwo

The length of 'VersionTwo' is 10 characters

Type a string (q - to quit):

The length of '' is 0 characters

Type a string (q - to quit):

q

\*\*\* End of Measuring String \*\*\*

# Manipulating

\*\*\* Start of Concatenating Strings Demo \*\*\*

Type the 1st string (q - to quit):

q

\*\*\* End of Concatenating Strings Demo \*\*\*

\*\*\* Start of Comparing String Demo \*\*\*

Type the 1st string to compare (q - to quit):

Version2

Type the 2nd String to compare:

Version2

'Version2' string is equal to 'Version2'

Type the 1st string to compare (q - to quit):

Computer

Type the 2nd String to compare:

Programming

'Computer' string is less than 'Programming'

Type the 1st string to compare (q - to quit):

Programming

Type the 2nd String to compare:

Computer

'Programming' string is greater than 'Computer'

Type the 1st string to compare (q - to quit):

987

Type the 2nd String to compare:

65

'987' string is greater than '65'

Type the 1st string to compare (q - to quit):

$%^

Type the 2nd String to compare:

!!!

'$%^' string is greater than '!!!'

'Programming' string is less than 'Test'

Type the 1st string to compare (q - to quit):

q

\*\*\* End of Comparing String Demo \*\*\*

# Converting

\*\*\* Start of Converting Strings to int Demo \*\*\*

Type an int numeric string (q - to quit):

q

\*\*\* End of Converting Strings to int Demo \*\*\*

\*\*\* Start of Converting Strings to double Demo \*\*\*

Type the double numeric string (q - to quit):

654.321

Converted number is 654.321000

Type the double numeric string (q - to quit):

-123.456

Converted number is -123.456000

Type the double numeric string (q - to quit):

4.9089

Converted number is 4.908900

Type the double numeric string (q - to quit):

101

Converted number is 101.000000

Type the double numeric string (q - to quit):

600

Converted number is 600.000000

Type the double numeric string (q - to quit):

Converted number is 0.000000

Type the double numeric string (q - to quit):

q

\*\*\* End of Conerting String to double Demo \*\*\*

# Tokenizing

\*\*\* Start of Tokenizing Words Demo \*\*\*

Type a few words separated by space (q - to quit):

q

\*\*\* End of Tokenizing Words Demo \*\*\*

\*\*\* Start of Tokenizing Phrase Demo \*\*\*

Type a few phrases separated by comma (q - to quit):

Seneca College

Phrase #1 is 'Seneca College'

Type a few phrases separated by comma (q - to quit):

Hello, World, CPR

Phrase #1 is 'Hello'

Phrase #2 is ' World'

Phrase #3 is ' CPR'

Type a few phrases separated by comma (q - to quit):

1, 2, 3, 4, 5

Phrase #1 is '1'

Phrase #2 is ' 2'

Phrase #3 is ' 3'

Phrase #4 is ' 4'

Phrase #5 is ' 5'

Type a few phrases separated by comma (q - to quit):

Version 1, Version 2

Phrase #1 is 'Version 1'

Phrase #2 is ' Version 2'

Type a few phrases separated by comma (q - to quit):

Hello! World!, CPR 101

Phrase #1 is 'Hello! World!'

Phrase #2 is ' CPR 101'

Type a few phrases separated by comma (q - to quit):

Type a few phrases separated by comma (q - to quit):

CPR, 101; Final

Phrase #1 is 'CPR'

Phrase #2 is ' 101; Final'

Type a few phrases separated by comma (q - to quit):

q

\*\*\* End of Tokenizing Phrases Demo \*\*\*