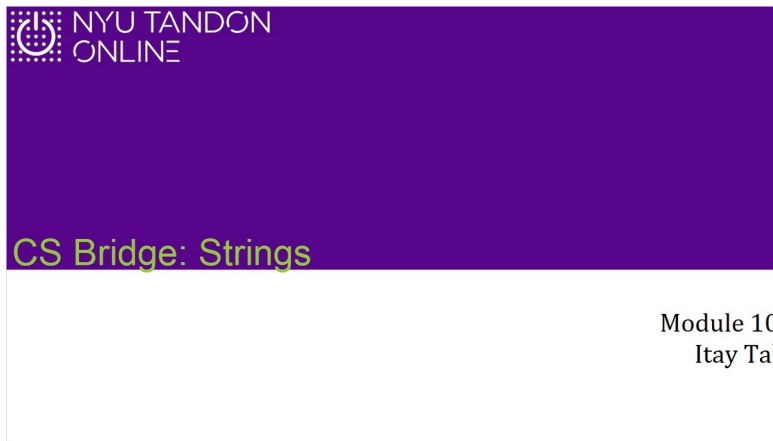


# CS Bridge Module 10 Strings

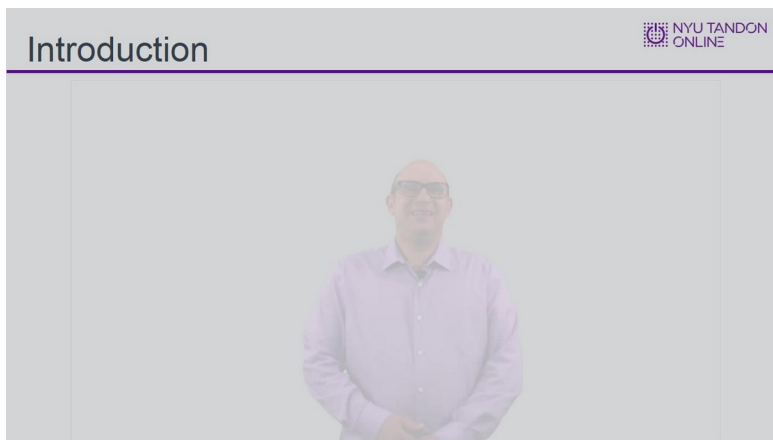
## 1. Strings

### 1.1 CS Bridge: Strings



Notes:

### 1.2 Introduction



Notes:

## 1.3 Initializing Strings

Initializing & Concatenating Strings

NYU TANDON  
ONLINE


```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string str;

    str = "abc";
    cout<<str + "def";

    return 0;
}
```

abcdef



Notes:

## 1.4 Reading Strings


Reading Strings

NYU TANDON  
ONLINE

```
int main() {
    • Initializing: string str; =
    • Concatenating: +
    cout<<"Please enter your name"<<endl;
    • Reading: cin>>str;
    getline(cin, str);
    cout<<str<<endl;

    return 0;
}
```

Please enter your name:  
Donald Duck  
Donald Duck



## 1.5 Knowledge Check

(Multiple Choice, 10 points, 1 attempt permitted)

## Knowledge Check



When reading in an unknown string, which line of code should you use?

- ☐ `cin >> str`
- ☒ `getline(cin, str)`
- ☐ `cout<<str`
- ☐ `getLine(str, cin)`

Correct	Choice	Feedback
	<code>cin &gt;&gt; str</code>	If the string is more than one word, only the first word will be read in, and the rest will be lost.
X	<code>getline(cin, str)</code>	Correct! <code>getline</code> allows you to read in a string regardless of how many spaces are in it
	<code>cout&lt;&lt;str</code>	Not quite! <code>cout</code> is how to print a string, not how to read in a string!
	<code>getLine(str, cin)</code>	This is not a valid function call. Note that the names of standard C++ functions are all in lowercase.

## cin >> str (Slide Layer)

Knowledge Check

NYU TANDON  
ONLINE

When reading in an unknown string, which line of code should you use?

Incorrect

If the string is more than one word, only the first word will be read in, and the rest will be lost.

Continue

☐ cin >> str

☒ getline(cin, str)

☐ cout<<str

☐ getLine(str,cin)

## getline(cin, str) (Slide Layer)

Knowledge Check

NYU TANDON  
ONLINE

When reading in an unknown string, which line of code should you use?

Correct

Correct! getline allows you to read in a string regardless of how many spaces are in it

Continue

☐ cin >> str

☒ getline(cin, str)

☐ cout<<str

☐ getLine(str,cin)

## cout<<str (Slide Layer)

Knowledge Check

NYU TANDON  
ONLINE

When reading in an unknown string, which line of code should you use?

Incorrect

Not quite! cout is how to print a string, not how to read in a string!

Continue

☐ cin >> str

☒ getline(cin, str)

☐ cout<<str

☐ getLine(str,cin)

## getline(str,cin) (Slide Layer)

NYU TANDON  
ONLINE

Knowledge Check

When reading in an unknown string, which line of code should you use?

Incorrect

This is not a valid function call. Note that the names of standard C++ functions are all in lowercase.

Continue

☐ cin >> str

☒ getline(cin, str)

☐ cout<<str

☐ getLine(str,cin)

## 1.6 Indexing Strings

NYU TANDON  
ONLINE

Indexing Strings

```
int main() {
    string str1, str2;
    char ch;
    Concatenating: +
    cout<<str1[0]<<" "<<str1[1]<<" "<<str1[2]<<endl;
    str1 = "abcde";
    getline(cin, str);
    cout<<str[5]<<endl;
    ch = str[3];
    return 0;
}
```

a b c

## 1.7 Slicing Strings

NYU TANDON  
ONLINE

Slicing Strings

```
int main() {
    string str1, str2;
    char ch;
    Concatenating: +
    str1 = "abcde";
    cout<<str1[0]<<" "<<str1[1]<<" "<<str1[2]<<endl;
    ch = str1[3];
    Indexing: str[0]
    cout<<str.substr(3, 2) << endl;
    str2 = str1.substr(2, 3);
    Slicing: str.substr(ind, len)
    cout<<str2<<endl;
    return 0;
}
```

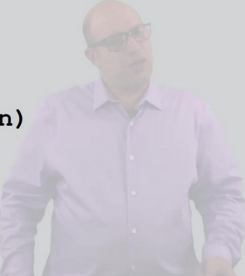
a b c  
de  
cde

## 1.8 Length

Length

NYU TANDON ONLINE

- **Initializing:** `=`
- **Concatenating:** `+`
- **Reading:** `cin`  
`getline(cin, str)`
- **Indexing:** `str[i]`
- **Slicing:** `str.substr(ind, len)`
- **Length:** `str.length()`



## 1.9 Printing Backwards Introduction

Printing Backwards Introduction

NYU TANDON ONLINE

**Problem**


Write a program that reads the user's name and prints it in a reverse order.

**Example**

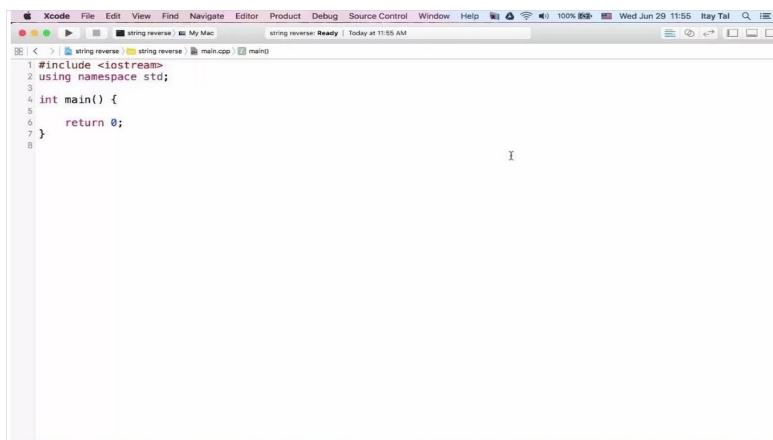
Please enter your name:

Donald Duck

kcuD dlanoD




## 1.10 Printing Backwards Implementation



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5
6     return 0;
7 }
8
```

## 1.11 Comparing Strings

Comparing Strings & Lexicographical Order




```
int main() {
    string str1, str2;
    // Concatenation: +
    str1 = "abc";
    str2 = "defg";
    // Reading from cin
    if (str1 < str2) {
        cout << str1 << " is smaller than " << str2 << endl;
    }
    // Writing to cout
    else {
        cout << str1 << " is not smaller than " << str2 << endl;
    }
    // Substring
    str1 = "abcde";
    str2 = "xyz";
    // Length
    cout << str1.length() << endl;
    cout << str2.length() << endl;
    // Comparing: ==, !=, <, >, <=, >=
    // Comparing:
    cout << "abc is smaller than de" << endl;
}
```

Notes:

## 1.12 First Word Introduction

First Word Introduction



Problem

Write a program that reads from the user 3 words and prints the one that comes first in an alphabetical order.

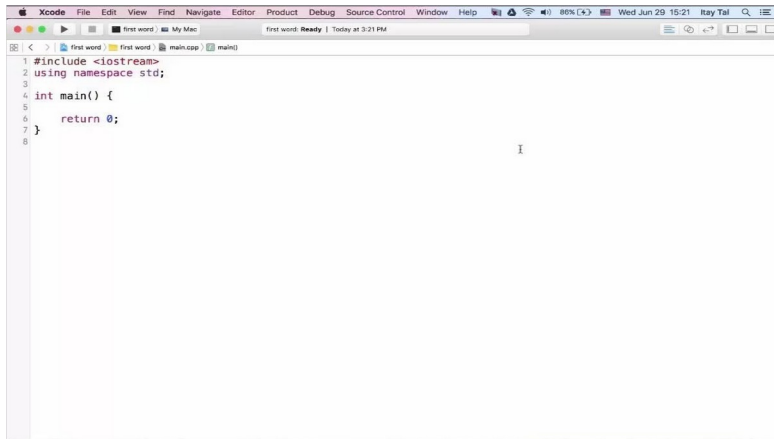
Example

Please enter 3 words, separated by a space:

dolphin cat tiger

cat


## 1.13 First Word Walkthrough



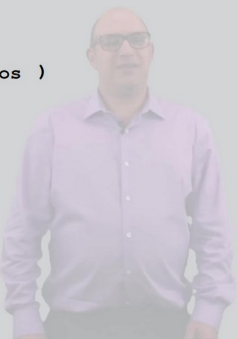
```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5
6     return 0;
7 }
8
```

## 1.14 Searching in a String

### Searching in a String




```
int main() {
    string str = "abcdbcdefg";
    cout<<str.find("de")<<endl;
    if (str.find("xyz") == string::npos)
        cout<<"Not found" <<endl;
    else
        cout<<str[i] <<endl;
    cout<<str.substr(ind, len)
    cout<<str.find(str, ind, len)
    cout<<str.length()
    return 0;
}
Comparing: ==, !=, <, >, <=, >=
Searching: str.find(s)
6
Not found
2
```




## 1.15 Starting Index

### Searching in a String from Starting Index



```
int main() {
    string str = "abcdbcdefg";
    cout<<str.find("de")<<endl;
    if (str.find("xyz") == string::npos)
        cout<<"Not found" <<endl;
    else
        cout<<str[i] <<endl;
    cout<<str.substr(ind, len)
    cout<<str.find("cd", 3);
    cout<<str.length()
    return 0;
}
Comparing: ==, !=, <, >, <=, >=
Not found
Searching: str.find(s)
2
5 str.find(s, startInd)
```





## 1.16 Knowledge Check

(Matching Drag-and-Drop, 10 points, unlimited attempts permitted)

NYU TANDON  
ONLINE

### Knowledge Check

Match the symbol or function with its corresponding purpose (in the context of strings).

Initializing	=
Concatenating	+
Reading	getline(cin,str)
Indexing	str[i]
Slicing	str.substr.(ind,len)
Length	str.length()
Comparing	==
Searching	str.find(s,startInd)

Correct	Choice
Initializing	=
Concatenating	+
Reading	getline(cin,str)
Indexing	str[i]
Slicing	str.substr.(ind,len)
Length	str.length()
Comparing	==
Searching	str.find(s,startInd)

### Feedback when correct:

That's right! You selected the correct response.

## Correct (Slide Layer)

NYU TANDON  
ONLINE

### Knowledge Check

Match the symbol or function with its corresponding purpose (in the context of strings).

Initializing	=
Concatenating	str.length()
Reading	==
Indexing	str.find(s,startInd)
Slicing	
Length	
Comparing	
Searching	

Correct

That's right! You selected the correct response.

Continue

## Try Again (Slide Layer)

NYU TANDON  
ONLINE

### Knowledge Check

Match the symbol or function with its corresponding purpose (in the context of strings).

Initializing	=
Concatenating	str.length()
Reading	==
Indexing	str.find(s,startInd)
Slicing	
Length	
Comparing	
Searching	

Incorrect

That is incorrect. Please try again.

Try Again

## 1.17 End of Module

NYU TANDON  
ONLINE

## End of Module

Exit

