

# Computer Programming

Dr. Deepak B Phatak

Dr. Supratik Chakraborty

Department of Computer Science and Engineering  
IIT Bombay

Session: separating multiple words in a line

# Quick Recap and Overview

---



- We have studied representation of strings by using char type arrays
- We have seen how strings can be manipulated in C++
- We discussed how to separate first and last names in a single line
- We will see a generalized approach to solve such a problem

# Problem

---



- Read an input sentence containing multiple words, separated by one or more blank spaces.
- There could be blank spaces even at the beginning, or at the end
- Separate each word, storing it in appropriate array
- Print all words, including the count of words

# Approach to solve this problem

---

- We will read the input sentence in a char array `sentence[200]`
- We do not know the number of words in the given sentence
  - We cannot have individually named array for each word
- We will store the words in a two dimensional char array  
`char word[50][200];`
- Each row is used to store one word
  - A word can contain up to 200 characters
- We scan the given sentence, one character at a time

# A Sample Sentence

- The input sentence will end with '\n', which will be replaced by gets with a '\0'. For example

Hello    World    How   are   you   \0

word[0] [] = H e l l o \0

word[1] [] = W o r l d \0

word[2] [] = H o w \0

word[3] [] = a r e \0

word[4] [] = y o u \0

# Program Design



Hello    World    How   are   you   \0

- When we scan the sentence, using an index  $i$  which is initially 0
  - Ignore blank, continue scanning
  - Check for non-blank, put it in the current word,
    - increment `character_count` (maintained for each word)
    - If non-blank is followed by blank, current word ends
      - Insert '\0' in the current word, reset `character_count` to 0, increment `numwords`, and start next word

# Program to separate all words in a sentence



```
/*  
* Program that counts number of words and prints them  
* It validates all the spaces (before, after, and in between the sentence)  
* Finally, prints all the words and word count  
*/  
#include<iostream>  
#include<cstdio>  
using namespace std;  
int main()  
{
```

# Program ... Define Variables Get input



```
char sentence[200];  
char words[50][200];  
int i, character_count=0, length, numwords=0;  
cout<<"Enter a string" <<endl;  
gets(sentence);  
for(i=0;sentence[i]!='\0';i++){
```



# Program ... (Assemble Words)

```
if(sentence[i]==' ') continue;
if(sentence[i]!=' '){
    words[numwords][character_count]=sentence[i];
    character_count++;
    if(sentence[i+1]==' ' || sentence[i+1]=='\0'){
        words[numwords][character_count]='\0';
        numwords++;
        character_count=0;
    }
}
} // end of scan, for loop ends here
```

# Program ... (Print all the words)



```
cout<<"Number of words:"<< numwords <<"\n";
for(i=0;i<numwords;i++) {
    cout<<words[i]<<endl;
}
return 0;
}
```

# Summary

---



- We can now handle character strings in C++
- In real life, it is often required to handle strings which are part of a text file
  - C++ compiler itself, for example, has to process our programs, which are lines of text, containing many symbols
  - Data in a spread sheet can be saved as a text file in a CSV format (“Comma Separated Values” format)
  - A practice problem will illustrate such a requirement
- We also need to know how to handle files in C++

**These aspects will be covered in the second part of CS101X**