

Strings

- ❖ An array of characters terminated with the null character '\0'
- ❖ Single quotes used to represent characters
- ❖ Double quotes used to represent null terminated string
- ❖ String Declaration and Initialization:
 - ◆ `char szAnswer[5] = {'Y','e','s','\0'};`
 - ◆ `char szAnswer[5] = "Yes";`
 - ◆ `char szAnswer[5];`
`szAnswer = "Yes";`
 - ◆ `char szAnswer[] = "Yes";`
(Note: Actually has 4 elements)

szAnswer[0]	'Y'
szAnswer[1]	'e'
szAnswer[2]	's'
szAnswer[3]	'\0'
szAnswer[4]	unknown

Copyright © 2005 R.M. Laurie 1

String Output

- ❖ Strings are contained in a one-dimensional array
- ❖ Strings are the only type of array which can be used directly with cout
- ❖ Example
 - `char szAnswer[6] = "Yes";`
 - `cout << szAnswer;`
- ❖ Output characters until '\0' is encountered

szAnswer[0]	'Y'
szAnswer[1]	'e'
szAnswer[2]	's'
szAnswer[3]	'\0'
szAnswer[4]	unknown
szAnswer[5]	unknown

Copyright © 2005 R.M. Laurie 2

String Input

- ❖ Using `cin >> czAnswer;` // has drawbacks:
 - ◆ No bounds checking
 - ◆ Can't input strings with white space characters
- ❖ `cin.get(szAnswer, sizeof(szAnswer));`
 - ◆ szAnswer is String to store characters
 - ◆ `sizeof(szAnswer)` maximum string size including NULL ('\0') character
- ❖ `cin.get(cAns);`
 - ◆ Retrieves a single character
- ❖ `cin.ignore(100, '\n')`
 - ◆ Clears Input Stream Buffer
 - ◆ Must be used to clear '\n'

szAnswer[0]	'Y'
szAnswer[1]	'e'
szAnswer[2]	's'
szAnswer[3]	'\0'
szAnswer[4]	'r'

Copyright © 2005 R.M. Laurie 3

#include <cstring> Library Functions

- ❖ `int strlen(szString);`
 - ◆ String Length excluding NULL
- ❖ `strcmp(szString1, szString2);`
 - ◆ Compares two strings
- ❖ `strcpy(szString1, szString2);`
 - ◆ Copies szString2 to szString1
- ❖ `strcat(szString1, szString2);`
 - ◆ Appends szString2 to szString1
 - ◆ Make sure szString1 is large enough

Copyright © 2005 R.M. Laurie 4

String Input Example

```

1. #include <iostream>
2. #include <cstring>
3. using namespace std;
4. int main(void)
5. {
6.     char szQuestion[80], szAnswer[20];
7.
8.     cout << "Enter a Question\n>";
9.     cin.get(szQuestion, sizeof(szQuestion));
10.    cin.ignore(100, '\n');
11.
12.    cout << "Enter an Answer\n>";
13.    cin.get(szAnswer, sizeof(szAnswer));
14.    cin.ignore(100, '\n');
15.
16.    strcat(szQuestion, "? - ");
17.    strcat(szQuestion, szAnswer);
18.    cout << szQuestion;
19.    return(0);
20. }

```

```

Enter a Question
>What is the 2nd Planet
Enter an Answer
>The Planet Venus
What is the 2nd Planet? - The Planet Venus

```

Copyright © 2005 R.M. Laurie 5

Console Character Input/Output

- ❖ Next 4 slides are for advanced programmers
- ❖ conio.h functions provide keyboard detection
- ❖ Old style DOS console I/O has limited support
 - ◆ char getch() // Get character and don't display
 - ◆ char getche() // Get character and echo character
 - ◆ putchar(char) // Put a character on output display

```

1. #include <iostream>
2. #include <conio.h>
3. using namespace std;
4. int main(void)
5. {
6.     char cEntry;
7.     cout << "Enter a Character\n>";
8.     cEntry = getche();
9.     cout << " = "<< hex << "0x" << int(cEntry);
10.    return(0);
11. }

```

```

Enter a Character
>A = 0x41

```

Copyright © 2005 R.M. Laurie 6

String and Type Conversion Functions

- ❖ Character Manipulation Functions

#include <cctype>

- ◆ char toupper(char) // Converts char to uppercase
- ◆ char tolower(char) // Converts char to lowercase
- ◆ bool isupper(char) // Is char uppercase?
- ◆ bool islower(char) // Is char lowercase?
- ◆ bool isdigit(char) // Is char a numerical digit (0-9)?
- ◆ bool isalpha(char) // Is char a letter (A-Z, a-z)?

- ❖ String to Number Conversion Functions

#include <cstdlib>

- ◆ double atof(char[]) // Converts string to double
- ◆ double atoi(char[]) // Converts string to int

Copyright © 2005 R.M. Laurie 7

conio.h Yes/No Input Example

```

1. #include <iostream>
2. #include <cctype>
3. #include <conio.h>
4. using namespace std;
5. int main(void)
6. {
7.     char cEntry;
8.     cout << "Do you like programming?\n>";
9.     do
10.    {
11.        cEntry = getche();
12.        cEntry = tolower(cEntry);
13.        if(cEntry == 'y')
14.            cout << "\bYes, I am glad.";
15.        else if(cEntry == 'n')
16.            cout << "\bNo, You will like it if you study";
17.        else
18.            cout << "\a\b";
19.    }while(cEntry != 'y' && cEntry != 'n');
20.    cout << "\n\nDone";
21.    return(0);
22. }

```

```

Do you like programming?
>Yes, I am glad.

Done

```

conio.h Number Input Example

```

1. #include <iostream>
2. #include <cstdlib>
3. #include <conio.h>
4. using namespace std;
5. int main(void)
6. {
7.     unsigned int nI, nNumber;
8.     char cEntry, szDigits[7];
9.     cout << "Enter your identification number\n";
10.    nI=0;
11.    do
12.    {
13.        cEntry = getch();
14.        if(cEntry == 0x0D) // Enter key pressed?
15.            break;
16.        if(cEntry == 0x08) // Backspace key pressed?
17.        {
18.            cout << "\b\b"; // Backspace and clear
19.            nI--; // Decrement Counter
20.            continue; // Try again
21.        }
22.        if(!isdigit(cEntry)) // Test if NOT digit
23.        {
24.            cout << '\a'; // Ring bell
25.            continue; // Try again
26.        }
27.        putchar(cEntry); // Display character
28.        szDigits[nI++] = cEntry; // Assemble string
29.    }while(nI < sizeof(szDigits)-1);
30.    szDigits[nI] = '\0';
31.    nNumber = atoi(szDigits);
32.    cout << "\nString = " << szDigits << " Number = " << nNumber;
33.    cout << endl;
34.    return(0);
35. }
    
```

Enter your identification number
>00345
String = 00345 Number = 345

Arrays Containing Strings

- ❖ To create an array containing strings declare and initialize a 2-dimensional array
- ❖ Access each string using first dimension only

```

1. #include <iostream>
2. using namespace std;
3. int main(void)
4. {
5.     char szFirstName[4][6]={"Rich", "Wilma", "Pam", "Joe"};
6.     int nI;
7.     for(nI=0; nI < 4; nI++)
8.         cout << szFirstName[nI] << endl;
9.     cout << "\nDone\n";
10.    return 0;
11. }
    
```

	[0]	[1]	[2]	[3]	[4]	[5]
[0]	R	i	c	h	\0	
[1]	W	i	l	m	a	\0
[2]	P	a	m	\0		
[3]	J	o	e	\0		

Rich
Wilma
Pam
Joe

Done

Copyright © 2005 R.M. Laurie 10

File Input and Output

- ❖ File Input reads text data from a text file
 - ◆ The text data can be assigned to variables
 - ◆ No prompting is necessary
 - ◆ Usually used for batch type processing
- ❖ File Output writes text data to a text file
 - ◆ Text strings can be written to a text file
 - ◆ Variable contents can be written to a text file
 - ◆ Usually used to record results or batch output
 - ◆ Use .txt or .dat file extension
 - ◆ File Output is required in last assignment

Copyright © 2005 R.M. Laurie 11

File Stream Input and Output

- ❖ File Input or Output requires use of library file: **#include <fstream>**
- ❖ Declare File Streams using file identifier:
 - ◆ **ifstream ifileDataIn;** // Declare input file stream
 - ◆ **ofstream ofileResults;** // Declare output file stream
- ❖ Open File
 - ◆ Open file for data input stream (Read from file)
ifileDataIn.open("Transactions.txt");
 - ◆ Open file for data output stream (Write to file)
ofileResults.open("Results.txt");
 - ◆ **ofileResults.open("Results.txt", ios::app);**
ofileResults.open("Results.txt", ios::trunc); //default
 - ◆ **if(!ifileDataIn)** // Evaluates true if file not opened
 - ◆ Use to display error message if file fails to open

Copyright © 2005 R.M. Laurie 12

File Read and Write

❖ Reading Data from File:

- ◆ Use input operator >>
- ◆ Instead of standard input `cin` use file identifier
- ◆ `ifstreamDataIn >> fPrice;`
- ◆ `ifstreamDataIn >> nQuantity >> fCost >> nYear;`

❖ Writing Data to File:

- ◆ Use output operator <<
- ◆ Instead of standard output `cout` use file identifier
- ◆ `ofstreamResults << nBlackChp << ' '`
- ◆ `ofstreamResults << "Blue = " << nBlueChp << endl;`
- ◆ `ofstreamResults.flush(); // Write to file now`

❖ For CS225 do NOT read and write to same file

Copyright © 2005 R.M. Laurie 13

Example 1: Simple File Write

```
#include <iostream>
#include <fstream>
using namespace std;
int main(void)
{
    int nI;
    ofstream ofileLoop;
    ofileLoop.open("Loop.txt", ios::trunc);
    if (!ofileLoop)
    {
        cout << "Can't Open Input File.\n";
        return 1;
    }
    ofileLoop << "This is a Simple Test\n";
    for(nI=1; nI <= 10; nI++)
        ofileLoop << nI << ' ';
    cout << "Done\n";
    ofileLoop.close();
    return 0;
}
```

```
This is a Simple
Test
1 2 3 4 5 6 7 8 9 10
```

Copyright © 2005 R.M. Laurie 14

Example 2: File Read

```
1. #include <iostream>
2. #include <fstream>
3. using namespace std;
4. int main(void)
5. {
6.     int nNum1, nNum2, nI;
7.     char cLetter;
8.     ifstream ifileData;
9.     ifileData.open("DataIn.txt");
10.    if (!ifileData)
11.    {
12.        cout << "Can't Open Input File.\n";
13.        return 1;
14.    }
15.    ifileData >> nNum1;
16.    ifileData >> nNum2;
17.    cout << "1st Number = " << nNum1;
18.    cout << "\n2nd Number = " << nNum2;
19.    cout << "\n      Sum = " << nNum1+nNum2 << endl << endl;
20.    for(nI=0; nI < 5; nI++)
21.    {
22.        ifileData >> cLetter;
23.        cout << cLetter << " = " << hex << int(cLetter) << endl;
24.    }
25.    cout << "\nDone\n\n";
26.    ifileData.close();
27.    return 0;
28. }
```

```
123  456
Aa B b C c D dEeF f
```

```
1st Number = 123
2nd Number = 456
      Sum = 579
```

```
A = 41
a = 61
B = 42
b = 62
C = 43
```

```
Done
```

Example 3: File Write (Append)

```
1. #include <iostream>
2. #include <fstream>
3. #include <iomanip>
4. using namespace std;
5. int main(void)
6. {
7.     int nEntry1, nEntry2;
8.     ofstream ofileResult;
9.     ofileResult.open("A:\\Results.txt", ios::app);
10.    if (!ofileResult.is_open())
11.    {
12.        cout << "Can't Open File.\n";
13.        return 1;
14.    }
15.    cout << "Enter 1st Number\n";
16.    cin >> nEntry1;
17.    cout << "Enter 2nd Number\n";
18.    cin >> nEntry2;
19.    cout << "Sum = " << nEntry1 + nEntry2 << endl;
20.    ofileResult << setw(4) << nEntry1 << " + "
21.        << setw(4) << nEntry2
22.        << " = " << setw(5) << nEntry1 + nEntry2 << endl;
23.    cout << "\nDone\n";
24.    ofileResult.close();
25.    return 0;
26. }
```

```
Enter 1st Number
>3
Enter 2nd Number
>4
Sum = 7
Done
```

```
3 + 5 = 8
64 + 37 = 101
123 + 456 = 579
1245 + 3456 = 4701
3 + 4 = 7
```