CL1004 – Object-Oriented Programming Lab

Lab # 2

Strings and File handling

Instructor: Muhammad Saad Rashad

Email: saad.rashad@nu.edu.pk

Department of Computer Science,

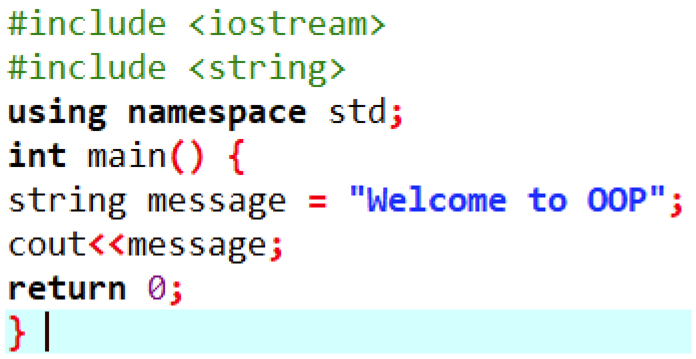
National University of Computer and Emerging Sciences FAST

Peshawar

1. C++ Strings:

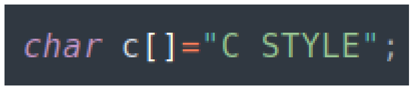
- Strings are used for storing text.
- A string variable contains a collection of characters surrounded by double quotes:
- Example: Create a variable of type string and assign it a value: string greeting = "Hello";
- To use strings, you must include an additional header file in the source code, the library Example //Include the string library #include // Create a string variable string message = "Welcome to OOP";

Example 1



```
#include <iostream>
#include <string>
using namespace std;
int main() {
string message = "Welcome to OOP";
cout<<message;
return 0;
}
```

ii. C-style strings:
Example 2:



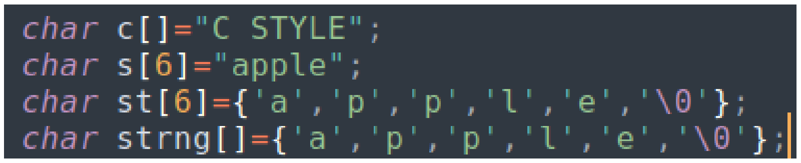
```
char c[]="C STYLE";
```

- When a compiler encounters a sequence of characters enclosed in #double quotation marks, it appends a character **'\0'** at the end by default.
-
- **Why do**

C		S	T	Y	L	E	\0

we include the \0 character at the end? This is known as the "null terminating character", and must be included when creating strings using this method. It tells C++/C that this is the end of the string.

a. String Declaration and Initialization:
Example 3:



```
char c[]="C STYLE";
char s[6]="apple";
char st[6]={'a','p','p','l','e','\0'};
char strng[]={'a','p','p','l','e','\0'};
```

2. **File Handling:**
   In C++, files are an integral part of data storage and manipulation. Unlike running programs where data is only temporary, file handling allows data to persist beyond program execution. This is crucial in the software industry, where applications need to save data for future use, such as configuration settings, logs, or user information.

File handling in C++ refers to the process of creating, reading, writing, and manipulating files within the file system using features provided by the C++ Standard Library. The <fstream> header is commonly used for file handling, providing three main classes:

- ifstream (input file stream): Used to read data from files.
- ofstream (output file stream): Used to write data to files.

- fstream (file stream): A combination of both, used for reading and writing to files.

- **File Operations:**

  In C++, you can perform four major operations on files, either text or binary:
  1. Creating a new file

Files are automatically created when you open them for writing using ofstream.

Example 4:

```
void crt_func(){
   char file_name[20];
   cout<<"Enter file name: ";
   cin>>file_name;
ofstream outFile(file_name); // Creates and opens the file
   if (outFile.is_open()) {
      cout << "File created successfully!" <<endl;
   } else {
      cerr << "Failed to create file." << endl;
   }
   outFile.close(); // Close the file after creating
}
```

2. Opening an existing file
3. Closing a file
4. Reading from and writing information to a file

Example 5 : Writing information to an existing file

```
void wrt_file()
{
   ofstream outFile("example.txt"); // Opens the file for writing
   if (outFile.is_open()) {
      outFile << "Hello, File Handling in C++!" << endl; // Write to the file
      outFile.close(); // Close the file after writing
      cout << "Data written successfully!" << endl;
   } else {
      cerr << "Failed to open file for writing." <<endl;
   }

}
```

Example 6:

```
void read_file()
{

   ifstream inFile("example.txt"); // Opens the file for reading
   if (inFile.is_open()) {
      string line;
      while (getline(inFile, line)) { // Reads the file line by line
         cout << line <<endl; // Print the content
      }
      inFile.close(); // Close the file after reading
   } else {
      cerr << "Failed to open file for reading." <<endl;
   }

}
```

3. File Modes:

When opening a file, various modes can be specified:

- ios::in: Open for reading.
- ios::out: Open for writing (creates a new file or truncates an existing one).
- ios::app: Append mode; data is added at the end of the file.
- ios::ate: Open and move the pointer to the end of the file immediately

Example 7: Append mode

```
void file_mode()
{
   ofstream outFile("example.txt", std::ios::app); // Opens the file in append
mode
   if (outFile.is_open()) {
      outFile << "This is appended text." << std::endl; // Append data to the file
      outFile.close(); // Close the file after appending
      cout << "Data appended successfully!" << endl;
   } else {
      cerr << "Failed to open file for appending." << endl;
   }
}
```

Task:

1. Create a new file with the name “your_rollno” the name of file must be provided by the user

2. Write data in the file using data must be written like this

- **a. Name:**
- **b. Roll_no:**
- **c. Hobbies:**

3. Append Age at the end using int array;

4. Write a code that takes two string and check whether the two strings are reverse of each other