Strings

Prepared and presented by -
Naomi Padmore

Strings

A string is an
array of characters

- Must include header file
  - `#include<string>`
- This means string characters can be accessed just like an array element

String Declaration and Initialization

```cpp
//Strings
#include <iostream>
#include <string>

using std::cin;
using std::cout;
using std::endl;

int main(){

    std::string test; //Declaration

    std::string firstName = "James"; // Declaration and Initialization
    std::string lastName = "Bond";


    return 0;

}
```

Operations on Strings

- Adding strings

```
//String operations
#include <iostream>
#include <string> //must include this library to use STL strings

using std::cin;
using std::cout;
using std::endl;

int main(){

    std::string firstName = "James";
    std::string lastName = "Bond";

    // we can add strings together to create longer strings
    std::string fullName = firstName + " "+ lastName;

    cout << "My name is " << lastName<< ", " << fullName << endl;

    return 0;
}
```

More Operations on Strings

- .size()
- .empty()
- .append()
- .compare()

```
//Assuming using std::string is declared
string userName = " ";

cout << "Enter your full name: ";
getline(cin, userName); // getline is used when we want to read in a line of information which has spaces
cout << "\nHello "<<userName <<endl;

string bdaySong = "Happy Birthday to you";
string bdaySongExt;
string test;
string person1 = "Jon";
string person2 = "Jane";
string person3;

//checking the size of a string variable, i.e. number of characters within string including spaces
cout<< "\nThe number of characters in bdaySong string is: "<<bdaySong.size()<<endl;

//checking if a string  variable is empty -> 0 (false), 1 (true)
cout<<"\nIs string variable test empty: "<<test.empty()<<endl;

//appending to a string variable
bdaySongExt = bdaySongExt.append(bdaySong + ", "+ bdaySong);
cout<<"\nExtended Birthday Song: "<<"\n"<<bdaySongExt<<endl;

//comparing strings -> 1 (less than), 0 (equal to), 1 (greater than)
cout<<"\nIs person1 = to person2? "<<person1.compare(person2)<<endl;

//assigning a string value to another string variable
person2 = person2.assign(person1);
cout<<"\nIs person1 = to person2 now? "<<person1.compare(person2)<<endl;

//assigning a sub string value to another string variable
person2 = "Jane";
person3 = person2.assign(person2,1,3);   // format -> .assign(string, starting index, # of characters)
cout<<"\nWelcome "<<person3<<"!"<<endl;
```

More Operations on Strings Cont'd

- .assign()
- .find()
- .insert()
- .erase()

```
//finding the index value of the first instance of a character/string occuring in the string
//-1 is returned if not present
// format -> .find(string to look for, index to begin search)
cout<<"\nlocation of the word to in bdaySongExt string is at index: "<<bdaySongExt.find("to",0)<<endl;
cout<<"location of the word to in person1 string is at index: "<<person1.find("to",0)<<endl;

//inserting a chracter/string within a string at a specific position
// format -> .insert(index position to insert from, string)
person1 = person1.insert(3, " Snow");
cout<<"\nWho is "<<person1<<endl;

//erasing characters from a string
// format -> .erase(index to start erasing from, number of chracters to erase)
person1 = person1.erase(3, 5);
cout<<"\nWho is "<<person1<<endl;
```

For More String Functions ->

- http://www.cplusplus.com/reference/cstring/

That’s All