C++ string

The string Class in C++

- C++ has a <string> library
- Include it in your programs when you wish to use strings: #include <string>
- In this library, a class **string** is defined and implemented
- It is very convenient and makes string processing easier than in C

Declaration of strings

- The following instructions are all equivalent. They declare s1 to be an object of type string, and assign the string “high school” to it:

```
string s1;
    s1=“high school”;
string s2= “high school”;
string s3(“high school”);
```

In c++ there is new class called string. Its an improvement on C-strings. It removes some of the limitations of C-Strings.

- There is no need to create the array of the right size to hold string variables.

Contd…….

- The String class assumes all the responsibility for memory management.
- It allows the use of overloaded operators,so we can concatinate string objects with **+** operator.

Eg:- s3=s1+s2

- It is more efficient and easy to use than C-strings.

```
#include<iostream>
#include<string>
using namespace std;
main()
{
    string s1("man");
    string s2=“hi";
    string s3;
    s3=s1;
    cout<<"s3="<<s3<<endl;

    s3="neither " + s1 + "nor";
    s3 += s2;
    cout<<"s3="<<s3<<endl;
    s1.swap(s2);
    cout<<s1<< "nor" <<s2 <<endl;
    return 0;
    }
```

Manipulating String Objects

- insert()
- erase()
- replace()
- append()
- length()
- at()
- find()
- substr()
- compare()

Manipulating String Objects

- string s1(“12345”);
- string s2(“abcde”);
- s1.insert(4, s2); // s1 = 1234abcde5

- s1.erase(4, 5); // s1 = 12345

- s2.replace(1, 3, s1); // s2 = a12345e
- s2.append(s1,2,3); //s2=a12345e345

```
#include<iostream>
#include<string>
using namespace std;
main()
{
   string s1("12345");
   string s2("abcde");

   s1.append(s2);
   cout<<s1<<endl;

   s1.append(s2,1,2);
   cout<<s1;

   return 0;
}
```

```
#include<iostream>
#include<string>
using namespace std;
main()
{
    string s1("12345");
    string s2("abcde");
    cout<<s1<<"  "<<s2<<endl;

    s1.insert(4, s2);
    cout<<s1<<endl;

    s1.erase(4, 5);
    cout<<s1<<endl;

    s2.replace(1, 3, s1);
    cout<<s2<<endl;
    return 0;
}
```

String Characteristics

Function	Task
size()	Number of elements currently stored
length()	Number of elements currently stored
max_size()	Maximum size of a string object that a system can support
empty()	Return true or 1 if the string is empty otherwise returns false or 0
swap()	Used to swap two string

String Characteristics

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str="welcome";
     cout << "Size = " << str.size() << endl;
     cout << "Length = " << str.length() << endl;
     cout << "Max Size = " << str.max_size() << endl;
     cout << "Empty:  "<< (str.empty() ? "yes" : "no") << endl;
    return 0;
}
```

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
   string str=“This is a c++ programming";
            cout<< "bytes = " <<sizeof(str)<<endl;
            cout << "Size = " << str.size() << endl;
            cout << "Length = " << str.length() << endl;
            cout << "Max Size = " << str.max_size() << endl;
            cout << "Empty:  "<< (str.empty() ? "yes" : "no") << endl;
   return 0;
}
```

compare

string s1(“ABC”);

string s2(“XYZ”);

int x = s1.**compare**(s2);

- x == 0 if s1 == s2
- x > 0 if s1 > s2
- x < 0 if s1 < s2

There is another overloaded version of compare

int compare(int start_1, int length_1, string s_2, int start_2, int length_2)

string s1, s2;

int x = s1.compare(0, 2, s2, 2, 2);

Program

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
   string s1 = "bcme";
   string s2 = "abcrome";
   cout<<s1.compare(s2)<<endl;
   cout<<s1.compare(2,2,s2,5,2);
   return 0;
}
```

```
#include<iostream>
#include<string>
using namespace std;
int main()
{  string str1="welcome";
   string str2="welldone";
   int x=str1.compare(str2);
   if(x==0)
   cout<<“Strings are same";
   else
   cout<<“Strings are different";
   cout<<endl<<str1.compare(0,3,str2, 0,3);


   return 0;
```

```
int main ()
{ string str1 ("green apple");
  string str2 ("red apple");
  if (str1.compare(str2) != 0)
    cout << str1 << " is not " << str2 << '\n';
  if (str1.compare(6,5,"apple") == 0)
    cout << "still, " << str1 << " is an apple\n";
  if (str1.compare(6,5,str2,4,5) == 0)
    cout << "therefore, both are apples\n";
  return 0;
}
```

Accessing Characters in Strings

Function	Task
at()	For accessing individual characters
substr()	For retrieving a substring
find() rfind()	For finding a specific substring For finding a specific substring from right
find_first_of()	For finding the location of first occurrence of the specific character(s)
find_last_of()	For finding the location of last occurrence of the specific character(s)
getline()	For taking the string object value from user standred io function

find() and substr()

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


int main()
{
   string s1 = "arlcome";
   string s2 = "wearomea";
   cout<<s1.find("lc")<<endl;
   cout<<s2.rfind(“ea”)<<endl;
   cout<<s2.find_last_of('e');

   return 0;
}
```

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


int main()
{
   string s1 = "welcome";
   string s2;
   s2=s1.substr(0,2);
   cout<<s2;
   return 0;
}
```

```
int main ()
{
  string str1 ("lovely professional university");

cout<< str1.find("ve")<<endl;
  cout<< str1.rfind("ve")<<endl<<endl;

  cout<<str1.find_first_of('l')<<endl;
  cout<<str1.find_last_of('l')<<endl<<endl;

}
```

find() and substr()

```
int main () {
 string str1 ("lovely professional university");
 string str2 ("ve");
 cout<<str1<<endl;
int x =str1.find(str2);
cout<<x<<endl;

string temp = str1.substr(x + str2.size() , 30 );
 cout<<temp<<endl;

int y= temp.find(str2);
 cout<<y;   // or cout<<y + x + str2.size();
}
```

getline

- string s;
- cout<<“Enter string”;
- getline(cin,s);

What is the difference between unsigned int length() and unsigned int size()?

a) Returns a different value

b) They give same result

c) Returns a different value but they are same

d) Returns a length

```
#include <iostream>
using namespace std;
int main() {
   string str1 ("The only way to do great work is to love what you do“);
   string str2 ("work“);
   unsigned found = str1.find(str2);
   cout << found << "\n";
   return 0;
}
```

a) 20
b) 23
c) 25
d) 21

- Choose the correct answer:

    1. char str1[] = {‘H’, ’e’, ’l’, ’l’, ’o’, ’\0’};
    2. char str2[]=“Hello”;
    3. string str3(“hello”);

A. str1 is a char array and str2 and str3 are strings.
B. str1 and str2 are char arrays and str3 is a string.
C. str1,str2 and str3, they all are strings.
D. None of the above