Traditional way

- Using null-terminated character arrays are not technically data types
- So C++ operators cannot be applied to them

- char s1[10],s2[10];

*test.cpp

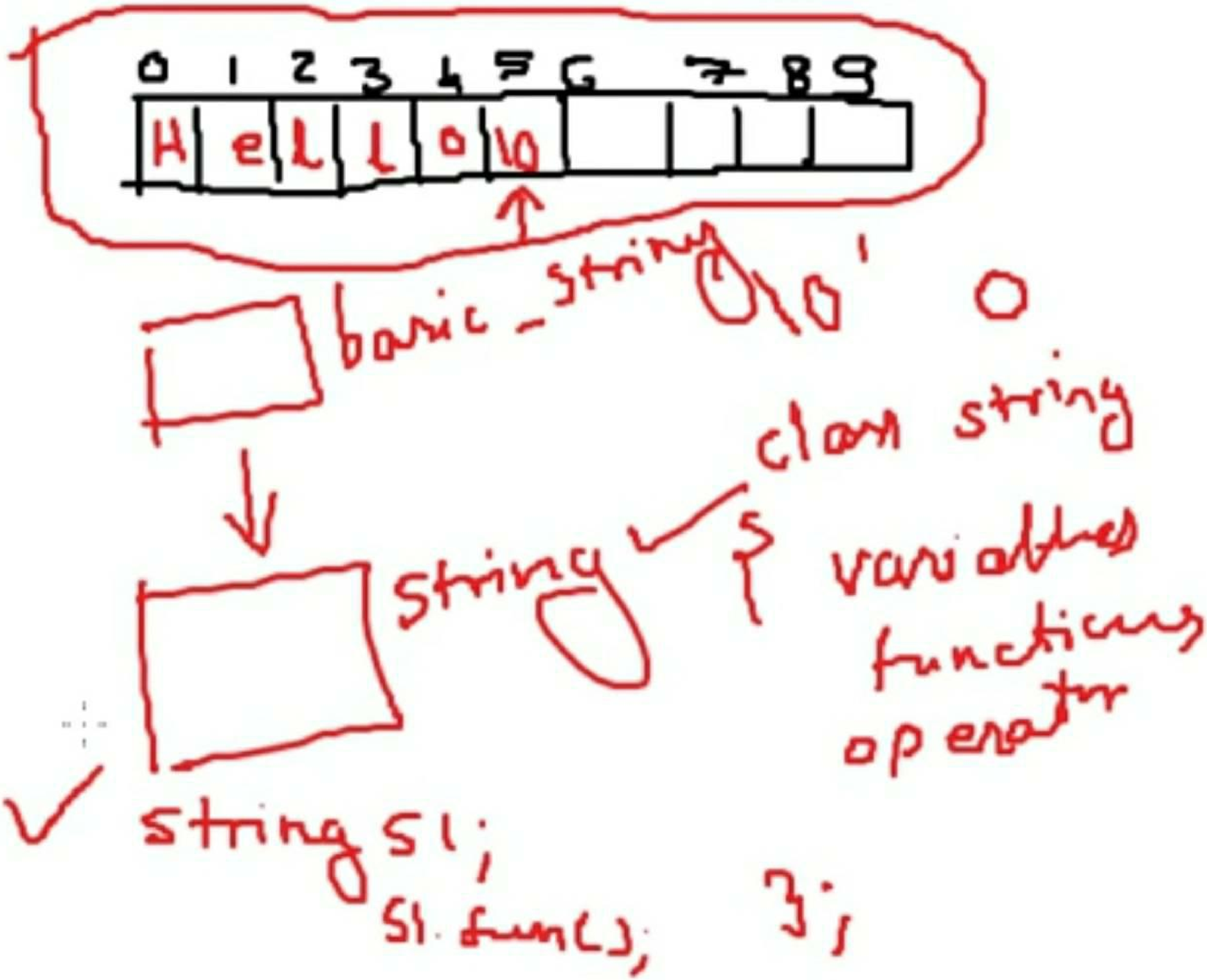
```cpp
#include<iostream>
using namespace std;
int main()
{
    char s1[10]="Hello";
    s1="Students"; //wrong
    strcpy(s1,"Students");
    char s2[20];
    s2=s1; //wrong
    strcpy(s2,s1);
    s2>s1; //wrong
    int i=strcmp(s2,s1);
    char s3[30];
    s3=s1+s2; //wrong
    strcpy(s3,strcat(s1,s2));


}

```

string class

- The string class is a specialization of a more general template class called basic_string
- Since defining a class in C++ is creating a new data type, string is derived data type
- This means operators can be overloaded for the class



string is safe than char array

- Careless programmer can overrun the end of an array that holds a null terminated string.
- For example, using strcpy()

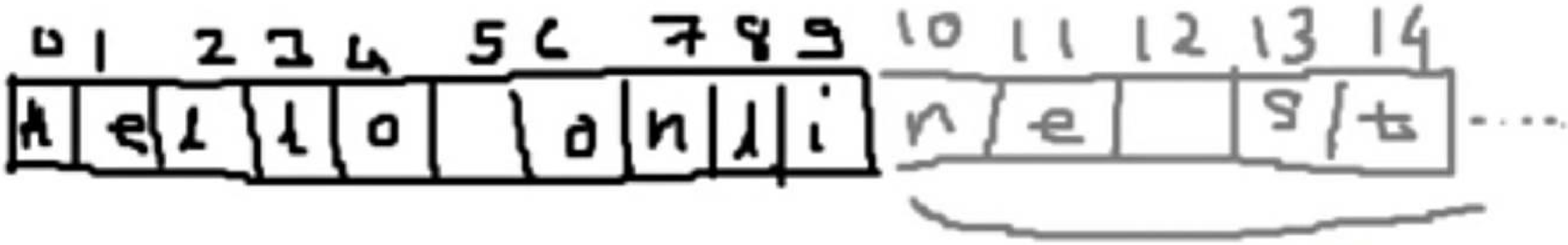
- string class handles such issues

string is in STL

- string is an another container class
- To use string class, you have to include string header
  - #include<string>

Constructors

- string class supports many constructor, some of them are
  - string()
  - string(const char *str)
  - string (const string &str)



S1

String S2 = "Hello";

String S2("Hello");

String S2 = String("Hello");

test.cpp

```cpp
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string s1;
    char str[]="Hello";
    string s2=str;
    string s3=s2;
}
```

Operators

- = Assignment
- + Concatenation
- += Concatenation assignment
- == Equality
- != Inequality
- < Less than
- <= Less than or equal
- > Greater than
- >= Greater than of equal
- [ ] subscripting

Insertion and extraction

- << Insertion (for output)
- >> Extraction (for input)

Mixed operations

- You can mix string objects with another string object or c style string

- C++ string can also be concatenated with character constant

Useful methods

- assign()
- append()
- insert()
- replace()
- erase()
- find()
- rfind()
- compare()
- c_str()
- size()