String, Input Commands, and Data Conversion

ACM-ICPC 2012 TUTORIAL #1

International College, KMITL: Gateway to professional success in the international arena

Content

- C-string and String in C++
- C/C++ Input Commands
- Data Conversion
- Reference: http://www.cplusplus.com

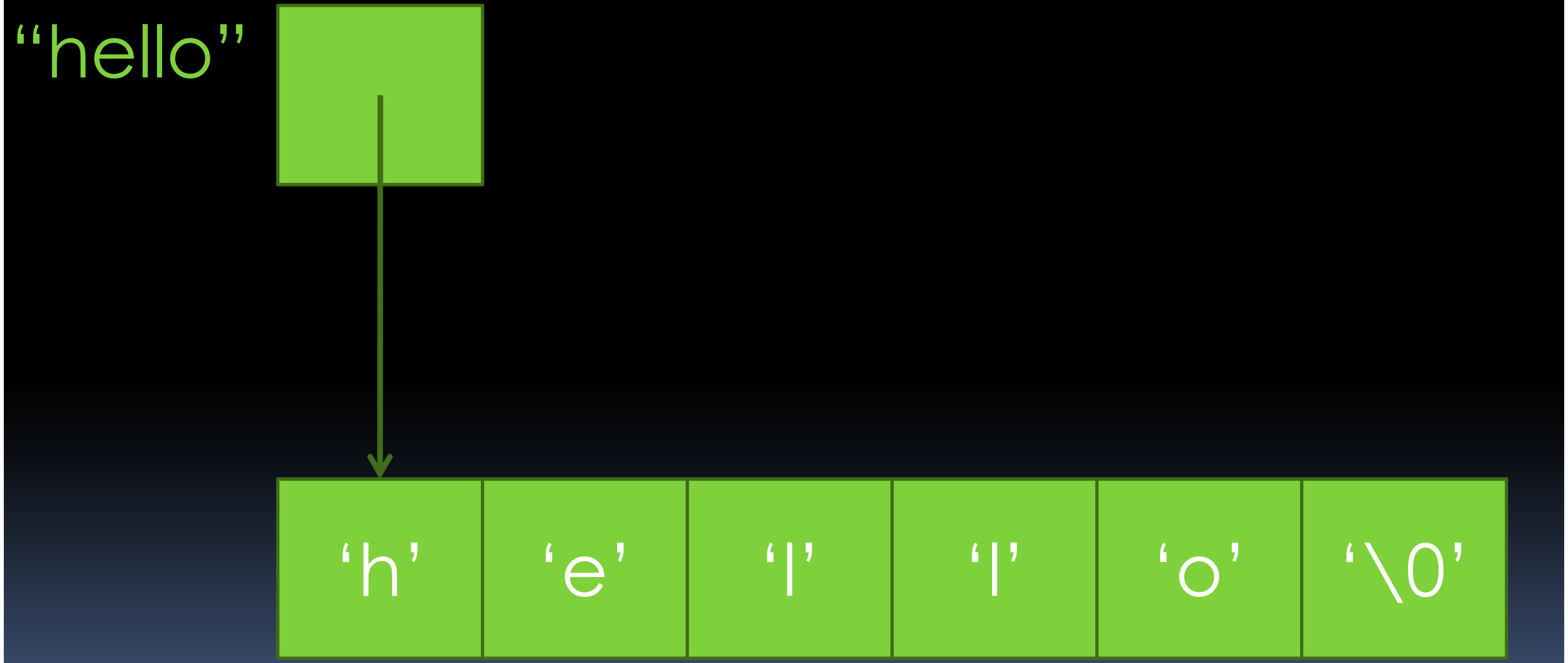
C-string and String in C++

C-string

- C-string (string in C language) is an array of characters
- Use \0 to signify the end
- The length of a C-string is the number of characters in the string (does not include \0).

- Example
  - char* str1;
  - char str2[256];

C-string



C-string related functions in <string.h>

Functions	Description
strlen()	Returns the number of characters
strcpy()	Assigns one C-string to another
strncpy()	Assign up to n characters of one string to another
strcat()	Appends one C-string to the other
strncat()	Appends up to n characters of one string to another
strcmp()	Compare two C-strings
strncmp()	Compare up to n characters from two strings
strchr()	Search for a certain character in a C-string

Problem with C-string

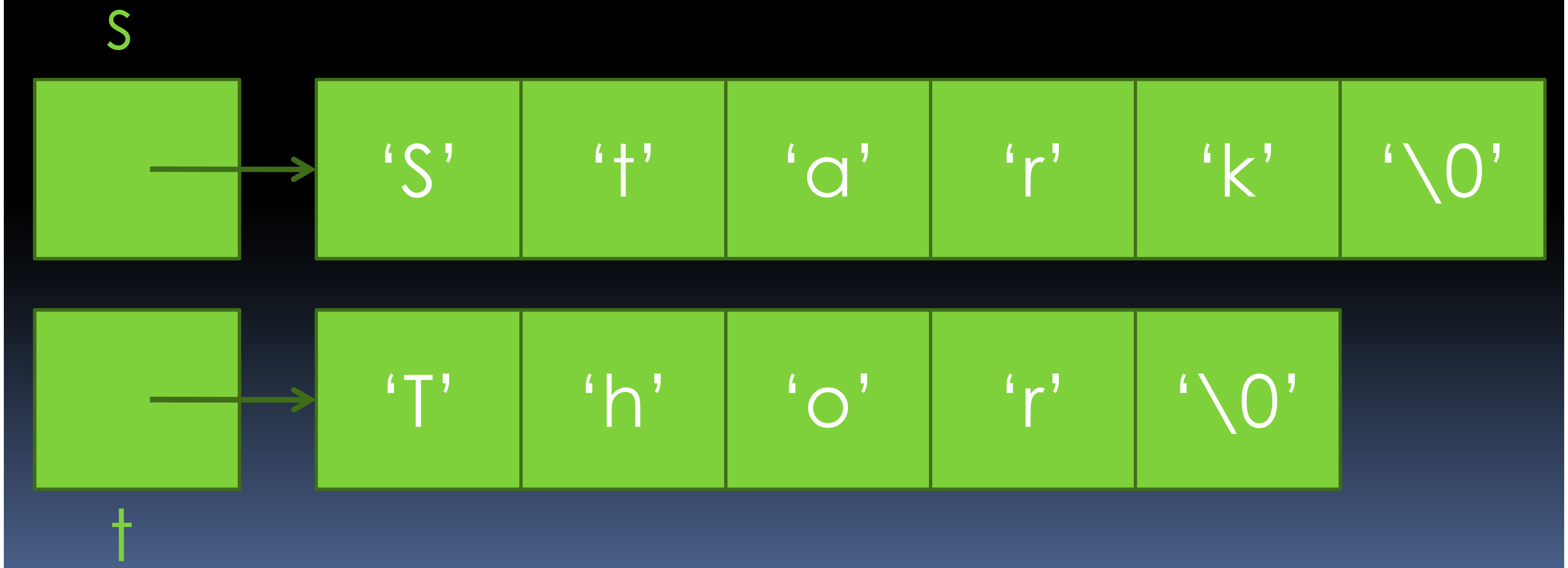
```
const char*s = “Stark” ;
const char*t = “Thor” ;

s = t ;        //CAUTION!!!
```

Problem with C-string

const char*s = “Stark” ;

const char*t = “Thor” ;



s

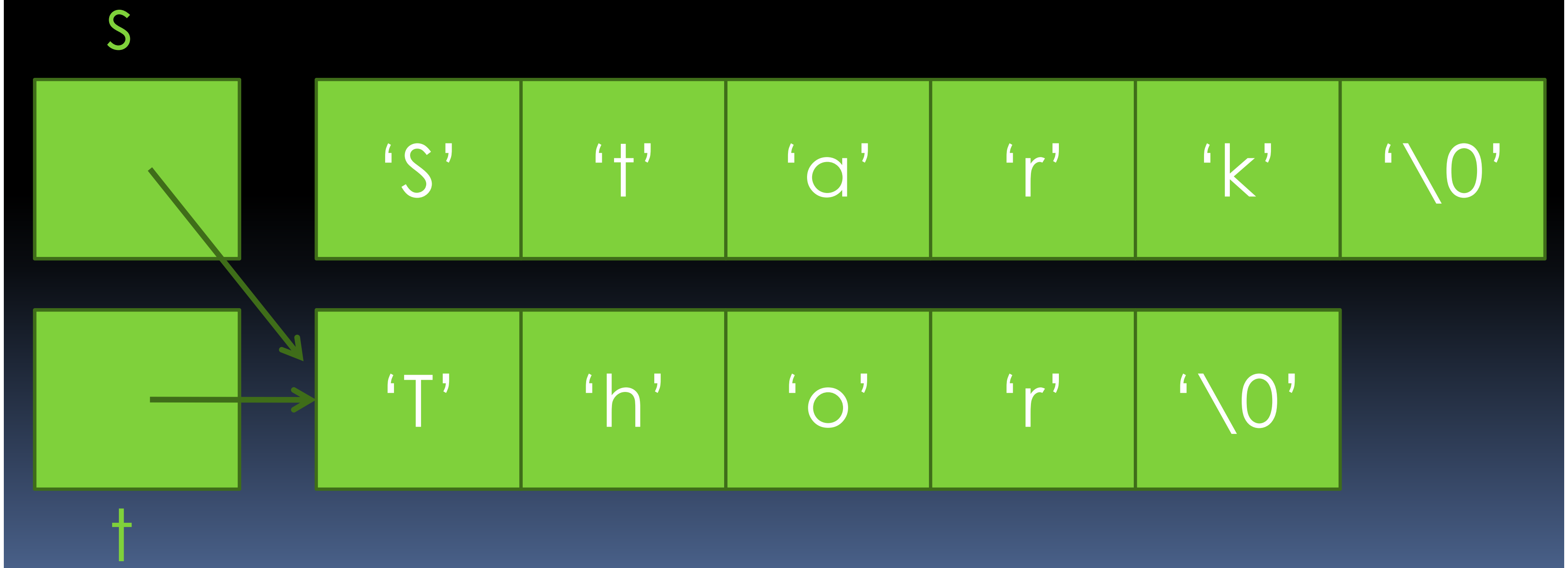
‘S’ ‘t’ ‘a’ ‘r’ ‘k’ ‘\0’

‘T’ ‘h’ ‘o’ ‘r’ ‘\0’

t

Problem with C-string

```
s = t ;        //pointer is assigned rather
               //the characters
```



Problem with C-string

- To copy C-string properly, the individual characters have to be copied.
- Must always ensure that sufficient memory is available.

String in C++

- include <string>   (not <string.h>)
- Use a class named string
- Need not worry about memory management
- Can use basic operators with string

Declare a string:	string s, t, text ;
Assign value:	s = “Hello Khun”;
Read string:	cin >> t ;
Compare empty string:	if(t == “ ”) return;
Assign string:	text = s + t ;
Display string:	cout << text;

Important methods of String

Methods (or operators)	Description
operator=	String assignment
operator==	String comparison
size	Return length of string
length	Return length of string
max_size	Return maximum size of string
resize	Resize string
capacity	Return size of allocated storage
clear	Clear string
empty	Test if string is empty

See the completed list at: http://www.cplusplus.com/reference/string/string/

Important methods of String

Methods (or operators)	Description
operator[]	Get character in string
at	Get character in string
operator+=	Append to string
append	Append to string
push_back	Append character to string
assign	Assign content to string
insert	Insert into string
erase	Erase characters from string
replace	Replace part of string
swap	Swap contents with another string

See the completed list at: http://www.cplusplus.com/reference/string/string/

Important methods of String

Methods (or operators)	Description
c_str	Get C-string equivalent
copy	Copy sequence of characters from string
find	Find content in string
find_first_of	Find character in string
find_last_of	Find character in string from the end
compare	Compare strings

See the completed list at: http://www.cplusplus.com/reference/string/string/

C/C++ Input Commands

cin

- Library
  - #include <iostream>
  - using namespace std;

- Example
  - int x, y, z;
  - cin >> x;
  - cin >> x >> y >> z;

```
// example on extraction
#include <iostream>
using namespace std;
int main () {
  int n;
  char str[10];
  cout << ″Enter a number: ″;
  cin >> n;
  cout << ″You have entered: ″ << n << endl;

  cout << ″Enter a hexadecimal number: ″;
  cin >> hex >> n;            // manipulator
  cout << ″Its decimal equivalent is: ″ << n << endl;

  cout << ″Enter a word: ″;
  cin.width (10);        // limit width
  cin >> str;
  cout << ″The first 9 chars of your word are: ″ << str << endl;

  return 0;
}
```

```
istream& getline ( istream& is, string& str, char delim );
istream& getline ( istream& is, string& str );
```

- Extracts characters from *is* and stores them into *str* until a delimitation character is found.
- For the second version, \n is used as the delimitation character .
- Library
  - #include <iostream>
  - #include <string>
  - using namespace std;

```
// getline with strings
#include <iostream>
#include <string>
using namespace std;

int main ()
{
  string str;
  cout << ”Please enter full name: ”;
  getline (cin,str);
  cout << ”Thank you, ” << str << ”.¥n”;
}
```

int scanf(char* format, &arg)

- Reads formatted data from stdin

- Library
  - #include <stdio.h>

- Example
  - int x;
  - float y;
  - char z;
  - scanf(“%d %f %c\n”, &x, &y, &z);

```
/* scanf example */
#include <stdio.h>
int main ()
{
  char str [80];
  int i;
  printf (″Enter your family name: ″);
  scanf (″%s″,str);
  printf (″Enter your age: ″);
  scanf (″%d″,&i);
  printf (″Mr. %s , %d years old.¥n″,str,i);
  printf (″Enter a hexadecimal number: ″);
  scanf (″%x″,&i);
  printf (″You have entered %#x (%d).¥n″,i,i);

  return 0;
}
```

int getchar(void)

- Returns the next character from the standard input (stdin).
- Library:
  - #include <stdio.h>
- Prototype
  - int getchar(void)
- Example

```
/* getchar example : typewriter */
#include <stdio.h>
int main ()
{
   char c;
   puts (”Enter text. Include a dot (’.’)
                      in a sentence to exit:”);
   do {
      c=getchar();
      putchar (c);
   } while (c != ’.’);
   return 0;
}
```

char* gets(char* s)

- Gets string from the standard input (stdin)
- Library
  - #include <stdio.h>

- Example
  - char str[256];
  - gets(str);

```
/* gets example */
#include <stdio.h>
int main()
{
  char string [256];
  printf (″Insert your full address: ″);
  gets (string);
  printf (″Your address is: %s¥n″,string);
  return 0;
}
```

Summary: C/C++ Input Commands

- iostream
  - cin
- iostream + string
  - getline

C++

- stdio.h
  - int scanf(char* format, &arg)
  - int getchar(void)
  - char* gets(char* s)

C

Data Conversion

C-string → Number

- #include <stdlib.h>

Methods (or operators)	Description
int atoi(char* s)	Convert s to int
int atol(char* s)	Convert s to long
double atof(char* s)	Convert s to double

- #include <stdio.h>

Methods (or operators)	Description
int sscanf(char* s, char* format, &args)	Equivalent to scanf() except that input is taken from the s string

Number → C-string

- #include <stdio.h>

Methods (or operators)	Description
sprintf(char* s, char* format, args);	Equivalent to printf() except that output is written to the s string. The string is terminated with a \o and must be large enough to hold the data.

String Object → Number

- #include <sstream>
- #inclulde <string>
- using namespace std;

```
template <typename T>
T StringToNumber ( const string &Text )
{
    istringstream ss(Text);
    T result;
    return ss >> result ? result : 0;
}
```

Number → String Object

- #include <sstream>
- #inclulde <string>
- using namespace std;

```
template <typename T>
string NumberToString ( T Number )
{
   ostringstream ss;
   ss << Number;
   return ss.str();
}
```