*Lecture #4*

*Strings and Structures*

* Strings are array of letters, pointers

Ex;

Print char- Printf(“Hello %c\n”, greet[0]);

Print str -Printf(“Hello %s\n”, greet);

Name a string

Str greet = “Hello”;

Char \*greet = “Hello”;

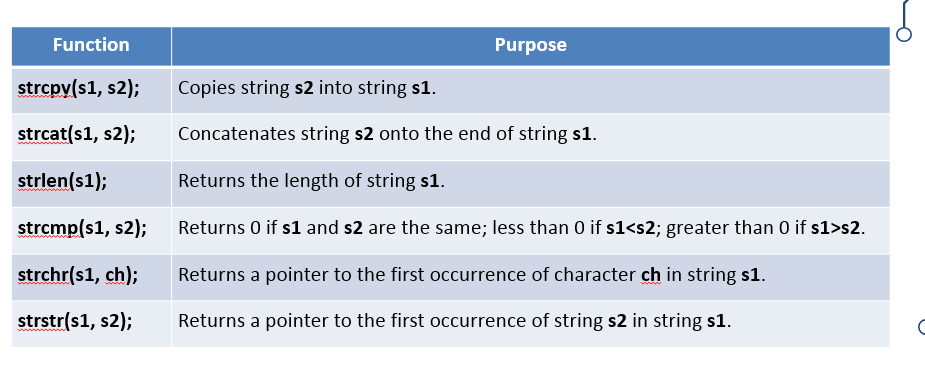
Char greet[#] = “word”;

Char greet[ ] = “word”;

1 2 3 4 5

W o r d \0

String library

STRCPY

* Source string must be shorter than destination string, or else string will ignore the difference
* Cannot assign lk this: str = string

Strlen

Returns

Printf(strlen(str))

Strcmp

* Returns if zero is identical
* Case sensitive
* Cant use ==
* Ex:
* Char letter []= “letter”
* Char word []= “letter”
* Word == letter; false, because points to memory addr

**Structures**

**struct [*structure tag*] {**

**member definition;**

**member definition;**

**member definition;**

**} [*one or more struct calls*];**

* **struct [struct\_name] x;**

**Acessing struct data**

* **Struct.value**
* **scanf("%f", &struct.value);**

**Struct arrays**

* **Can DO**

**Struct Functions**

* **[return type] function name (struct [struct name] x ) {}**

**Pointers to struct**

* **struct Books {**
* **… } book1;**
* **struct Books \*struct\_pointer;**
* **struct\_pointer = &book1;**

**Acess Pointer to struct**

* **(\*struct\_pointer).title;**
* **struct\_pointer->title;**
* **A struct cannot contain its own type, but it can contain a pointer to its own type**

**Unions**

**Easier structs**

**You learn enum later**

**Same thing as structs**

**Question**

**Still do not understand union**