Introduction to C – Delving further into the Blue Sea of Cs

Lec 9

Of Arrays, Input, Output Statements

Manta

The Manta Ray qualifies The Mirror Test

The manta ray has the largest brains of any fish.

Few animals, mostly primates, have passed this test used to determine self-awareness.

Dolphins, elephants, monkeys magpies, and even a robot (!), in some cases, can recognize themselves in the mirror. Humans, chimpanzees and orangutans are the only species for which the results are compelling & reproducible. This implies that self-awareness may be limited to humans and some great apes.
The mirror test may not be the litmus test for self-awareness in all animals.

https://www.newscientist.com/article/2081640-manta-rays-are-first-fish-to-recognise-themselves-in-a-mirror/#ixzz6piVGFqiF

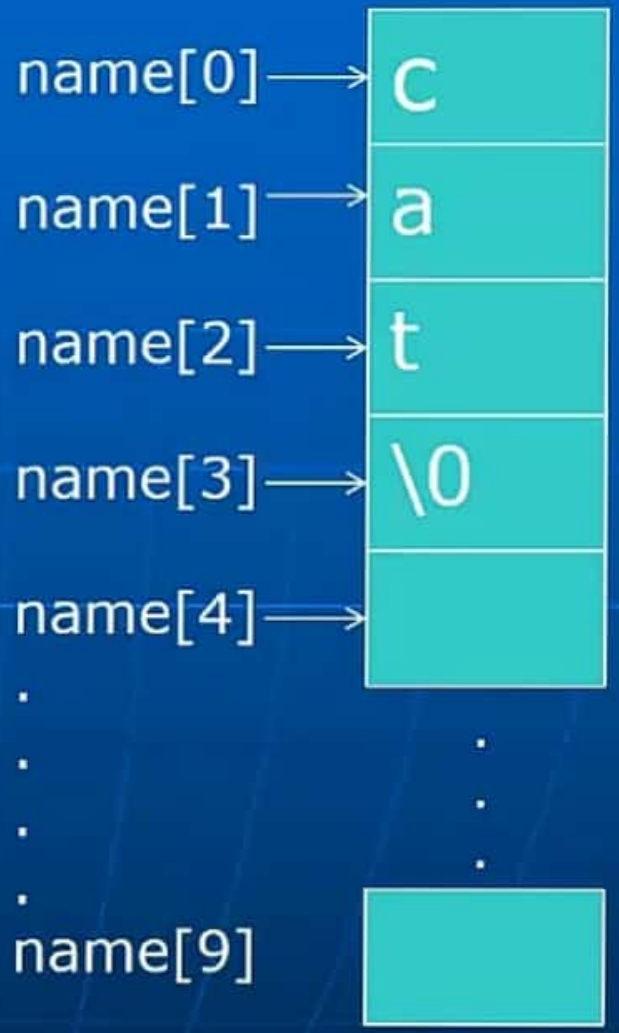
Character strings

- Array of characters where the last character is '\0' (null character)

- Eg:
  char name[10] = "cat";

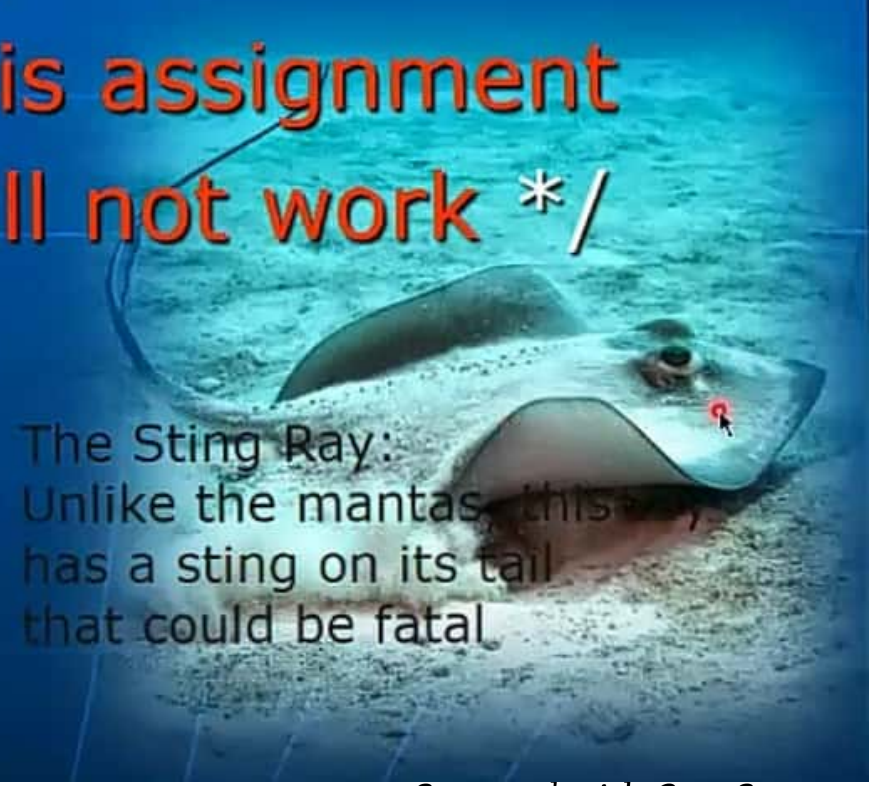
name[0] → 'c'
name[1] → 'a'
name[2] → 't'
name[3] → '\0'

Note:
name[4] to name[9] contains garbage.



Character strings …

- int i;
  char name[10];
  i = 123; /* OK */
  name = "cat"; /* this assignment will not work */



Character strings …

char name[10] = "cat"; / initialization*/*
char name[10];
name = "cat"; / wrong assignment*/*
assign each character as
name[0] = 'c'; name[1] = 'a'; … name[3] = '\0';

*OR one can use:

strcpy(name, "cat");
strcpy() is a library function in *string.h*
which means you need to
#include<string.h>
at the beginning of the program
(Check this function out)

Input and Output

Manta Ray: One of few animals that have an evolved filter feeding mechanism for **INPUT**. Let's see how we can do it in C!

- Input and output facilities are not part of the C language itself!

- These are supported by a set of library functions in *stdio.h*

getchar( ) and putchar( )

```
char c;                /* declaration */
c = getchar( );        /* reads a character
                          from keyboard
                          (stdin)*/

putchar(c);            /* writes the character
                          stored in the variable c
                          on the screen
                          (stdout)*/
```

scanf (formatted input)

```
int num;
char ch;
scanf("%d", &num); /* reads an
                      integer
                      from stdin into
                      num  */
scanf("%c", &ch);    /* reads a char
                      from stdin
                      into ch */
```

scanf (formatted input)

```
/* If multiple inputs need to be read
 using a single scanf */
scanf("%d  %c", &i, &ch);
```

Format specifiers

int variable

char variable

The Mysterious '&' in the Sea of Cs

- Every variable should have a location in the memory.
- If *i* is the name of a variable then *&i* is its address in the memory.
- *scanf* should be given the *addresses* of the locations where it should store the values which are being read.

The Mysterious & … in the Sea of Cs

- int i = 2;

  printf("%d %p\n", i, &i);

- 2 0x7FFF4604DFCC

Value as an integer

Address of the variable *i*

printf (formatted output)

```
int i; char ch;
i = 125;
ch = 'a';
printf("%d  %c \n", i, ch);
printf("%c\n%d",ch,i);
```

Output on the screen

```
125 a

a

125
```

printf

```
char ch = 'A';
printf("%d", ch);
```

The fish whose names begin with C within the Sea of Cs are proficient in C. So, let's ask the Clown fish what this program would output and why?

Recognize me?

Marlin from the Great Barrier Reef.
My son, Nemo, got lost when he ventured into the open C. (Finding Nemo!)

printf

```
char ch = 'A';
printf("%d", ch);
```

The fish w... Sea of Cs... vn f...

Aha! So you thought you could trick me, eh?

It will print **65** because you used %d to print the content of a character variable. NB: ASCII equivalent of A is 65

Recognize me?

Marlin from the Great Barrier Reef.
My son, Nemo, got lost when he ventured into the open C. (Finding Nemo!)

How do we read or write a float?

%f → float

```
float nemo;
scanf("%f ", &nemo);
printf("%f \n", nemo);
```

How do we read or write a string?

```
char str[64];
printf("What is your name?");
scanf("%s", str);
printf("Hello ... %s \n", str);
```

NB: *str* happens to be the starting address of the string i.e. str[0]. So it is already an address. Thus & is not required.

Output on screen

What is your name? Nemo

Hello … Nemo

-

Explore other ways of formatting

%c	Character	char unsigned char
%d	Signed Integer	short unsigned short int long
%e or %E	Scientific notation of float values	float double
%f	Floating point	float
%g or %G	Similar as %e or %E	float double

Strings: Points to Ponder

```
#include <stdio.h>
int main()
{
   char str[20];
   scanf("%s", str);
   printf("%s\n", str);
   return 0;
}
```

Hey Marlin, what would be the output if the input given is:

Hello there!

A Rare phenomenon of floating Cheese on the Curface of the Cea of Cs

More in the coming lecture