C Data Types, Arrays, Strings, and Files

CIS 308

Jorge Valenzuela

KANSAS STATE UNIVERSITY

1

C Programming Language

Key Words in C Language

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned

KANSAS STATE UNIVERSITY

2

C Programming Language

Key Words in C Language

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned

KANSAS STATE UNIVERSITY

3

C Data Types

Type	Size in bytes	Values	Format Specifier
void	?	?	-
int	?	?	%d
float	?	?	%f
double	?	?	%f
char	?	?	%c

KANSAS STATE UNIVERSITY

4

C Data Types

Type	Size in bytes	Values
void	?	None

KANSAS STATE UNIVERSITY

5

C Data Types

Integer Data Type

Type	CPU				Values
	8	16	32	64	
int	Bytes				?
	2	2	4	4	

KANSAS STATE UNIVERSITY

6

C Data Types

Integer Data Type **Variations**

Type	CPU				Values
	8	16	32	64	
int or signed int	Bytes				-32,768 to 32,767 -2,147,483,648 to 2,147,483,647
	2	2	4	4	
unsigned int	2	2	4	4	0 to 65,535 0 to 4,294,967,295
short int	2				-32,768 to 32,767
long int	4				-2,147,483,648 to 2,147,483,647

KANSAS STATE UNIVERSITY

7

C Data Types

Floating Point Data Type

Type	CPU				Values
	8	16	32	64	
float	4 Bytes				1.2E-38 to 3.4E+38 6 decimal places
double	8 Bytes				2.3E-308 to 1.7E+308 15 decimal places

The **IEEE Standard for Floating-Point Arithmetic** (**IEEE 754**)

KANSAS STATE UNIVERSITY

8

C Data Types

`char` Data Type

Type	Bytes	Values
char	1	-128 to 127 or 0 to 255 Depending on implementation
unsigned char	1	0 to 255
signed char	1	-128 to 127

KANSAS STATE UNIVERSITY

9

Extended Types

- Arrays
  - Declaration, initialization, arrays and functions
  - Multi-dimensional arrays
- Strings
  - String vars
  - String input and output
  - String Functions

KANSAS STATE UNIVERSITY

10

Arrays

- Declaration

```
type  name[size]

int intArray[10];

for (i = 0; i < 10; i++) {
  intArray[i] = 0;
}
```


KANSAS STATE UNIVERSITY

11

Arrays

- Array and Functions

```
void printArray(int[], int);
int main () {
   int intArray[10];
   for (int i = 0; i < 10; i++) {
     intArray[i] = i;
   }
   printArray(intArray, ?);
}
```


KANSAS STATE UNIVERSITY

12

Arrays

- Array and Functions

```
void printArray(int[], int);
int main () {
   int intArray[10];
   for (i = 0; i < 10; i++) {
     intArray[i] = i;
   }
   printArray(intArray, 10);
}
```


KANSAS STATE UNIVERSITY

13

Arrays

- Multi-Dimensional Array

```
int main () {
   char array[2][6];

   array[1][4]= 'B';
}
```

[0][0]					[0][5]
[1][0]					[1][5]

KANSAS STATE UNIVERSITY

14

Arrays

- Multi-Dimensional Array

```
int main () {
   char array[2][6];

   array[1][4]= 'B';
}
```

[0][0]					[0][5]
[1][0]				B	[1][5]

KANSAS STATE UNIVERSITY

15

Strings

- Declaration

```
char str[]= "Hello";
```

H	e	l	l	o	\0

```
char str[6] =
{'H', 'e', 'l', 'l', 'o', '\0'}
```


KANSAS STATE UNIVERSITY

16

Strings

- Initialization

```
char str1[] = "abc";
char str2[] = {'a', 'b',
'c'};(?)

str1 = str2; (?)

str1[3] = 'Z'; (?)
```


KANSAS STATE UNIVERSITY

17

Strings

- Initialization

```
char str1[] = "abc";
char str3[] = "abc";

str1 == str3; //
```


KANSAS STATE UNIVERSITY

18

Strings

- Initialization

```
char str1[4];
str1 = "abc"; //Illegal
```


KANSAS STATE UNIVERSITY

19

Strings

- Input and Output

```
char name[10];

printf("Enter your name: ");
scanf("%s", name); //(&?)
printf("Hello %s\n", name);
```

What if you enter "George Washington"? (18)

KANSAS STATE UNIVERSITY

20

Strings

- Input and Output

```
char[] fgets(char s[], int size, FILE *stream);
```

```
printf("Enter your name: ");
fgets(name, 10, stdin);
printf("Hello %s\n", name);
```


KANSAS STATE UNIVERSITY

21

Strings

- Conversions
  - `#include <stdlib.h>`
  - atoi: converts from a string to an int
  - atof: converts from a string to a double
  - itoa: converts from an int to a string
  - ftoa: converts from a double to a string

KANSAS STATE UNIVERSITY

22

Strings

- String Functions

```
#include <string.h>

char[] strcat(char str1[], char str2[]);
char[] strcat(char str1[], char str2[],
int lim) ;


int strcmp(char str1[], char str2[])
//It returns num:
// num <0, num == 0, or num >0
```


KANSAS STATE UNIVERSITY

23

Strings

- String Functions

```
#include <string.h>

int strlen(char str[])

char[] strtok(char str[], char delim[])
int strncmp(char str1[], char str2[], int n)
char[] strrchr(char str[], char c)
int strspn(char str1[], char str2[])
```


KANSAS STATE UNIVERSITY

24

File I/O

- Opening a file

```
#include <stdio.h>

FILE* fopen(char[] filename, char[] mode)
```

mode	Description
"r"	Open for reading (file must exist)
"w"	Open for writing (overwrites old data)
"a"	Open for appending (creates file if necessary)
"r+"	Open for reading and writing (file must exist)
"w+"	Open for reading and writing (overwrites old data)
"a+"	Open for reading and appending (opens at end of file)

KANSAS STATE UNIVERSITY

25

File I/O

- Opening a file

```
FILE *fp;
fp=fopen("c:\\test.txt", "r");

if(fp == 0) {
   printf("Error opening the
file\n");
   exit(0);
}
```


KANSAS STATE UNIVERSITY

26

File I/O

- Closing a file

```
#include <stdio.h>

int fclose(FILE* fp)
```

- Read from a file

```
int fscanf(FILE *stream, char str[],
variable addresses...)
```


KANSAS STATE UNIVERSITY

27

File I/O

- Read from a file

```
int fscanf(FILE *stream, char str[],
variable addresses...)

FILE *fp = fopen("data.txt", "r"); // Alice 22
char name[20];
int age;
if (fp != NULL) {
   while (fscanf(fp, "%s %d", name, &age) != EOF)
{
      printf("%s is %d years old\n", name, age);
   }
   fclose(fp);

}
```


KANSAS STATE UNIVERSITY

28

File I/O

- Read from a file

```
char[] fgets(char s[], int size, FILE
*stream)
```


KANSAS STATE UNIVERSITY

29

File I/O

- Write to a file

```
int fprintf(FILE* fp, char str[],
variables to print...)

File *fp
fp= fopen(c:\\test.txt", "w");
fprintf(fp, "HelloWOrld\n");
```


KANSAS STATE UNIVERSITY

30

C Programming Language

Key Words in C Language

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned

KANSAS STATE UNIVERSITY

31

Multiple Files

- Size
- Organization
- Reuse

KANSAS STATE UNIVERSITY

32

Multiple Files

- Function prototypes are placed in header files (.h files).
- Function themselves are implemented in a .c file.
- Include in your .c file the .h file that contains the function(s) you want to use.

KANSAS STATE UNIVERSITY

33

C Preprocessor

- C Preprocessor
  - A tool used to process a program prior to compiling it
  - It provides facilities for defining
    - Macros
    - File inclusion
    - Conditional compilation
  - It is automatically called when the C compiler is invoked

KANSAS STATE UNIVERSITY

34

C Preprocessor

- C Preprocessor
  - Include files

```
#include

#include <stdio.h>
#include <stlib.h>
#include <string.h>
#include “project2.h”
```


KANSAS STATE UNIVERSITY

35

C Preprocessor

- C Preprocessor
  - Conditional Compilation

```
#if constant-expression
#ifdef identifier
#ifndef identifier
#define identifier
#endif
```

In header files

```
#ifndef HEADER_FILENAME
#define HEADER_FILENAME
…
…
#endif
```


KANSAS STATE UNIVERSITY

36

C Preprocessor

- C Preprocessor
  - Macro
  - Allows an identifier to be associated with a text string
  - All occurrences of the identifier will be replaced with the associated string
  - Simple and parametrized
  - **Simple Macro**

```
#define name value

#define PI 3.14159
```


KANSAS STATE UNIVERSITY

37

C Preprocessor

- C Preprocessor
  - Simple Macro

```
#define PI 3.14159
#define TRUE 1  // Constant
#define FALSE 0 // Constant
```

  - **Parametrized Macro**

```
#define MAX(x, y) ((x)>(y) ? (x):(y))
#define SUM(a, b) a + b


int x = 3;
int y = 4;
int result = SUM(x, y);
```


KANSAS STATE UNIVERSITY

38

C Data Types

The explosion of the Ariane 5 (1996)
In 1996, Europe's newest and unmanned satellite-launching rocket, the Ariane 5, was blown up just seconds after taking off on its maiden flight from Kourou, French Guiana. The European Space Agency estimated that total development of Ariane 5 cost more than $8bn (£4bn). On board Ariane 5 was a $500 million (£240 million) set of four scientific satellites created to study how the Earth's magnetic field interacts with Solar Winds. According to a piece in the New York Times Magazine, the self-destruction was triggered by software trying to stuff "a 64-bit number into a 16-bit space."
"This shutdown occurred 36.7 seconds after launch, when the guidance system's own computer tried to convert one piece of data--the sideways velocity of the rocket--from a 64-bit format to a 16-bit format. The number was too big, and an overflow error resulted. When the guidance system shut down, it passed control to an identical, redundant unit, which was there to provide backup in case of just such a failure. But the second unit had failed in the identical manner a few milliseconds before. And why not? It was running the same software," the article stated.

KANSAS STATE UNIVERSITY

39

Arrays, Strings, and Files

Lab Activity

KANSAS STATE UNIVERSITY

40