

C Programming Lecture

1

ABOUT "C"

✓ *C programming language* - *Structured and disciplined approach to program design.*

- C is a structured programming language
- C supports functions that enables easy maintainability of code, by breaking large file into smaller modules
- Comments in C provides easy readability
- C is a powerful language.
- C programs built from
 - Variable and type declarations
 - Functions
 - Statements
 - Expressions

Basic Structure Of "C" Programs

```
#include<stdio.h>
#include<conio.h>

void main()
{
    -- other statements
}
```

Header Files

Entry Point Of Program

Indicates Starting of Program

C's Character Set

C does not use every character set and key found on modern computers . The only characters that C - Language uses for its programs are as follows:

- ✓ A-Z all alphabets
- ✓ a-z all alphabets
- ✓ 0-9
- ✓ # % & ! _ { } [] \$ % % & & & |
- ✓ space . , ; ' \$ "
- ✓ + - / * =

Some more basics

- Keywords
 - char, static, if, while, return etc.
- Data Types
 - int, char, float, etc.
 -
- Arithmetic Operators
 - + (Plus), - (Minus), * (Multiplication), /(Division)

5

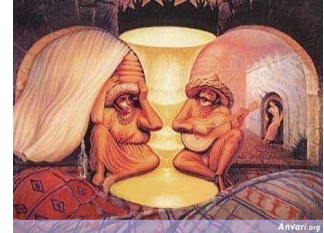
The keywords

- ✓ **"Keywords"** are words that have special meaning to the **C** compiler.
- ✓ Their meaning cannot be changed at any instance.
- ✓ Serve as basic building blocks for program statements.
- ✓ All keywords are written in **only lowercase.**

Keywords in Ansi "C"

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

What do you see?



Simple Data Types

Everything is Just a Bunch of Bits

- Bits can represent many different things
 - Depends on interpretation
- You and your program must keep track of what kind of data is at each location in the computer's memory
 - E.g., program data types

Simple Data Types

9

Big Picture

- Processor works with finite-sized data
- All data implemented as a sequence of bits
 - Bit = 0 or 1
 - Represents the level of an electrical charge
- Byte = 8 bits
- Word = largest data size handled by processor
 - 32 bits on most older computers
 - 64 bits on most new computers

Simple Data Types

Data types in C

- Only really four basic types:
 - char
 - int (short, long, long long, unsigned)
 - float
 - double

- Sizes of these types
- vary from one machine
- to another!

Type	Size (bytes)
char	1
int	4
short	2
long	8
long long	8
float	4
double	8

Simple Data Types

Characters (char)

- Roman alphabet, punctuation, digits, and other symbols:
 - Encoded within one byte (256 possible symbols)

- In C:

```
char a_char = 'a';
char newline_char = '\n';
char tab_char = '\t';
char backslash_char = '\\';
```

Simple Data Types

Arithmetic Operators in C

Name	Operator	Example
Addition	+	num1 + num2
Subtraction	-	initial - spent
Multiplication	*	fathoms * 6
Division	/	sum / count
Modulus	%	m % n

Arithmetic Operators Rules of Operator Precedence

Operator(s)	Precedence & Associativity
()	Evaluated first. If nested (embedded) , innermost first. If on same level, left to right.
* / %	Evaluated second. If there are several, evaluated left to right.
+ -	Evaluated third. If there are several, evaluated left to right.
=	Evaluated last, right to left.

Some more Data Types

- Primary : int, float, char
 - int (signed/unsigned)(2,4Bytes): used to store integers.
 - char (signed/unsigned)(1Byte): used to store characters
 - float, double(4,8Bytes): used to store a decimal number.
- User Defined:
 - typedef: used to rename a data type
 - typedef int integer; can use integer to declare an int.
 - enum, struct, union

Some more Arithmetic Operators

- Prefix Increment : ++a
 - example:


```
» int a=5;
» b=++a; // value of b=6; a=6;
```
- Postfix Increment: a++
 - example


```
» int a=5;
» b=a++; //value of b=5; a=6;
```

Contd...

- Modulus (remainder): %
 - example:


```
» 12%5 = 2;
```
 - Assignment by addition: +=
 - example:


```
» int a=4;
» a+=1; //(means a=a+1) value of a becomes 5
```
- Can use -, /, *, % also

Contd...

- Comparison Operators: <, >, <=, >=, !=, ==, !, &&, || .
 - example:


```
» int a=4, b=5;
» a<b returns a true(non zero number) value.
```
- Bitwise Operators: <<, >>, ~, &, |, ^ .
 - example


```
» int a=8;
» a= a>>1; // value of a becomes 4
```

Operator Precedence

- Meaning of $a + b * c$?
is it $a+(b*c)$ or $(a+b)*c$?
- All operators have precedence over each other
- $*$, $/$ have more precedence over $+$, $-$.
 - If both $*$, $/$ are used, associativity comes into picture. (more on this later)
 - example :
 - » $5+4*3 = 5+12 = 17$.

My first C program!

```
#include <stdio.h>
// program prints hello world
int main() {
    printf ("Hello world!");
    return 0;
}
```

Output: Hello world!

Example 1

```
#include <stdio.h>
// program prints a number of type int
int main() {
    int number = 4;
    printf ("Number is %d", number);
    return 0;
}
```

Output: Number is 4

Example 2

```
#include <stdio.h>
// program reads and prints the same thing
int main() {
    int number ;
    printf ("Enter a Number: ");
    scanf ("%d", &number);
    printf ("Number is %d\n", number);
    return 0;
}
```

Output : Enter a number: 4
Number is 4

more and more

```
#include <stdio.h>

int main() {
    /* this program adds
    two numbers */
    int a = 4; //first number
    int b = 5; //second number
    int answer = 0; //result
    answer = a + b;
}
```

Note

Errors

Compilation

Compiler generally gives the line number at which the error is present.

Run time

C programs are sequential making the debugging easier.

Home Work

- Meaning of
 - Syntax
 - Semantics of a programming language
- Find the Output:
 - `value=value++ + value++;`
 - `Value=++value + ++value;`
 - `value=value+++ ++value;`