PG – DESD

Module – Embedded C Programming

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Mobile No - 9890662093

Tokens

- C program is made up of functions.
- Function is made up of statements.
- Statement contain multiple tokens.
  - Keywords
  - Data Types
  - Identifiers
  - Variables
  - Constants
  - Operators

Keywords

- Keywords are predefined words used in program, which have special meanings to the compiler.
- They are reserved words, so cannot be used as identifier.
- K & R C has 27 keywords. C89 added 5 keywords. C99 added 5 new keywords.

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

Identifiers

- Identifiers give names to variables, functions, defined types and pre-processor macros.

- Rules of Identifiers:
  - Should start with alphabet or with _ (underscore)
  - Can include alphabets, _ (underscore),digits
  - Case sensitive

- Examples:
  - Var_1 //Valid
  - 1_var // Not Valid
  - _var //valid
  - Var-1 // invalid
  - Basic Salary //invalid

Data Types, Variables & Constants

- C allows computations to be performed on various types of data.
  - Numerical: Whole numbers, Real numbers
  - Character: Single character, Strings

- Fixed data values are said to be constants.
  - 12, -45, 0, 2.3, 76.9, 1.23456e+2, 'A', "Sunbeam", etc.

- Data is hold in memory locations identified by names called as variables.
  - Variable must be declared before its use in the program.
  - As per need, variable have some data type.

- Simple C data types are: int, double, char.
  - Data type represents amount of space assigned to the variable.
  - It also defines internal storage of the data.

Data Types, Variables & Constants

- Variable examples
  - int number = 10;
  - double basic_salary = 20000.0;
  - char letter = 'A';
  - int roll_number;
  - roll_number = 20;
  - double price = 200.0;
  - price = 300.0;
- Constant examples
  - -23, 1L, 34U, 3UL, 0x41, 0101,
  - 1.234f, 1.234567e+2, …
  - 'A', '\101', '\x41'
  - "SunBeam", "A\101\x41"
- Each variable is assigned some memory location.
- Size of data type of given variable or constant is found by sizeof() operator.

printf()

- Arbitrary strings and variable values can be printed using printf() function.
  - int - %d
  - double - %lf
  - char - %c
  - …

- Examples:
  - printf("Hello PreCAT @ Sunbeam");
  - printf("%d", roll_number);
  - printf("%d %lf %c", number, basic_salary, letter);
  - printf("Book price is %lf", price);

- Escape sequences
  - \n, \r, \t, \b, \a, \\, %%, \", \'

Data Types

- Data type defines storage space and format of variable.
- Primitive types
  - int
  - short
  - long
  - char
  - float
  - double
- Integer types can be signed/unsigned
- Derived types
  - Array
  - Pointer
  - Function
- Type qualifiers
  - const and volatile
- printf() format specifiers
  - %d, %u, %o, %x
  - %hd, %hu
  - %ld, %lu
  - %c
  - %f, %e
  - %lf
- User defined types
  - struct
  - union
  - enum
- void type – represent no value.

Data Types

- char
  - signed char (-128 to 127)
  - unsingned char (0 to 255)
- int / long
  - signed int (-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647)
  - unsinged int (0 to 65,535 or 0 to 4,294,967,295)
- short int
  - signed short (-32,768 to 32,767)
  - unsingned short(0 to 65,535)
- long long
  - signed long (-9223372036854775808 to 9223372036854775807)
  - unsinged long(0 to 18446744073709551615)
- float: ± 3.4E +/- 38
- double: ± 1.7E +/- 308

Data Types

C Basic Data Types	32-bit CPU		64-bit CPU	
	Size (bytes)	Range	Size (bytes)	Range
char	1	-128 to 127	1	-128 to 127
short	2	-32,768 to 32,767	2	-32,768 to 32,767
int	4	-2,147,483,648 to 2,147,483,647	4	-2,147,483,648 to 2,147,483,647
long	4	-2,147,483,648 to 2,147,483,647	8	-9,223,372,036,854,775,808-9,223,372,036,854,775,807
long long	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807
float	4	3.4E +/- 38	4	3.4E +/- 38
double	8	1.7E +/- 308	8	1.7E +/- 308

Thank you!

Devendra Dhande <devendra.dhande@sunbeaminfo.com>