PART II
THE C PROGRAMMING LANGUAGE

Advanced Program

Nguyen Thi Thu Huong - Soict - HUST

UNIT 1
AN OVERVIEW OF C LANGUAGE

Nguyen Thi Thu Huong - Soict - HUST

An Overview of C

- C History
- C programming environment
- C program structure

WHERE DID C COME FROM?

- Derived from the B language
- B was derived from the BCPL (Basic Combined Programming Language) language

C History

- C developed by Dennis Ritchie at AT&T Bell Labs in the 1970s.
  - Used to maintain UNIX systems
  - Many commercial applications written in c
- C++ developed by Bjarne Stroustrup at AT&T Bell Labs in the 1980s.
  - Overcame several shortcomings of C
  - Incorporated object oriented programming
  - C remains a subset of C++

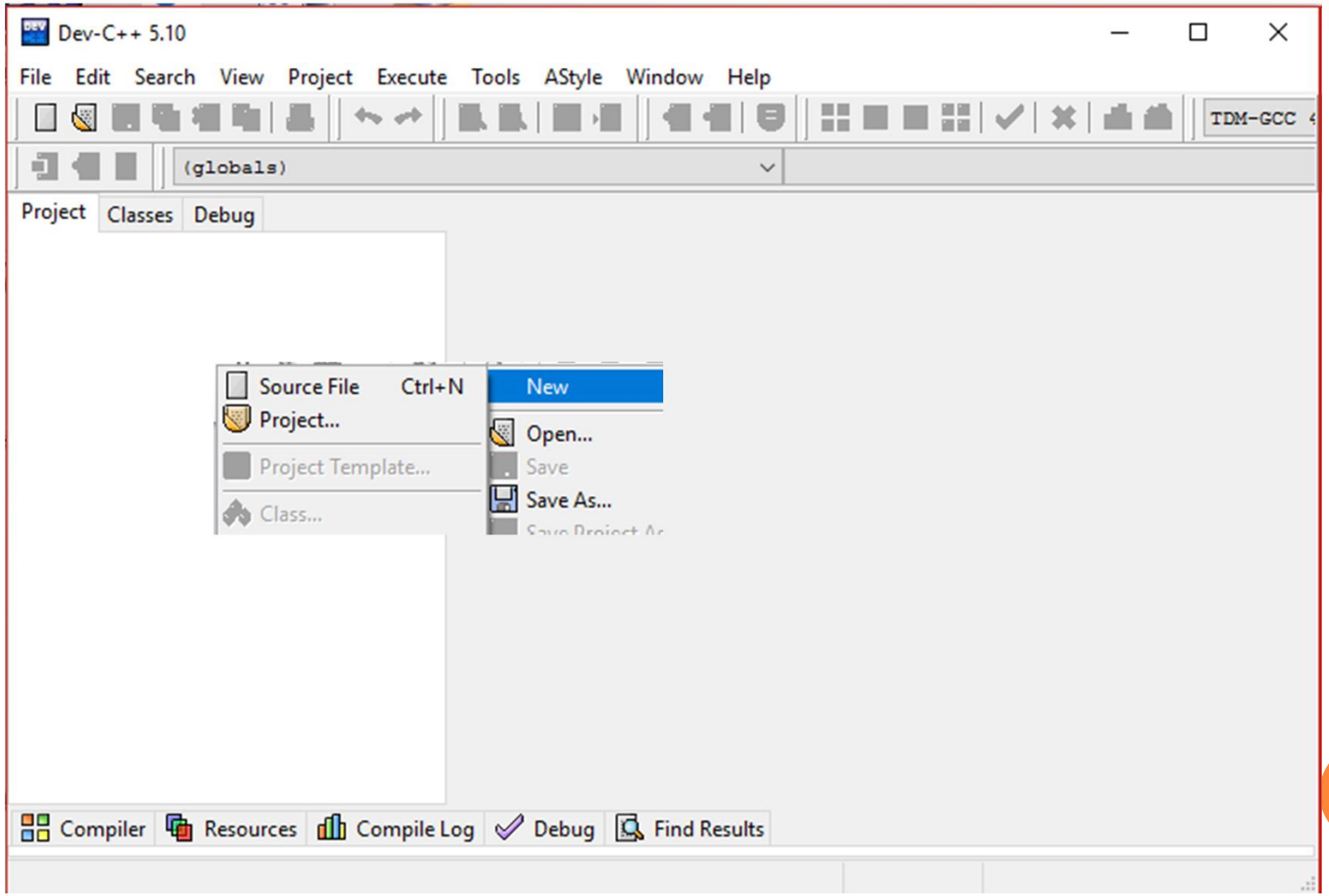
CHARACTERISTICS OF C

- Small size
- Extensive use of function calls
- Loose typing
- Structured language
- Low level (BitWise) programming readily available
- Pointer implementation - extensive use of pointers for memory, array, structures and functions.

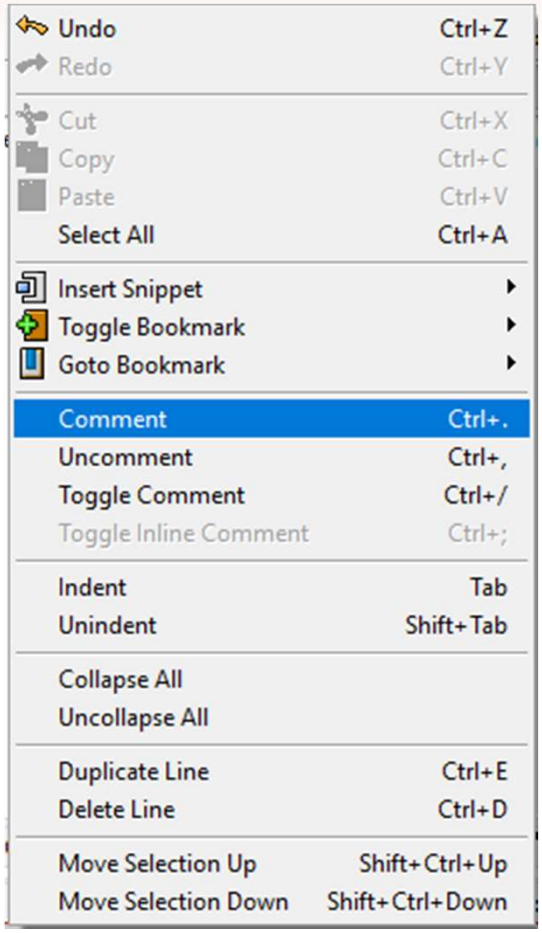
C vs. C++

- C is a subset of C++.
- All of features in C are contained in C++
- C++ adds more libraries with functions for object oriented programming
- C++ also adds more keywords and some added features.

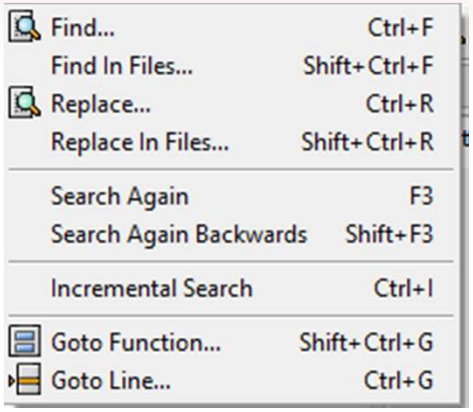
THE DEV C PROGRAMMING ENVIRONMENT

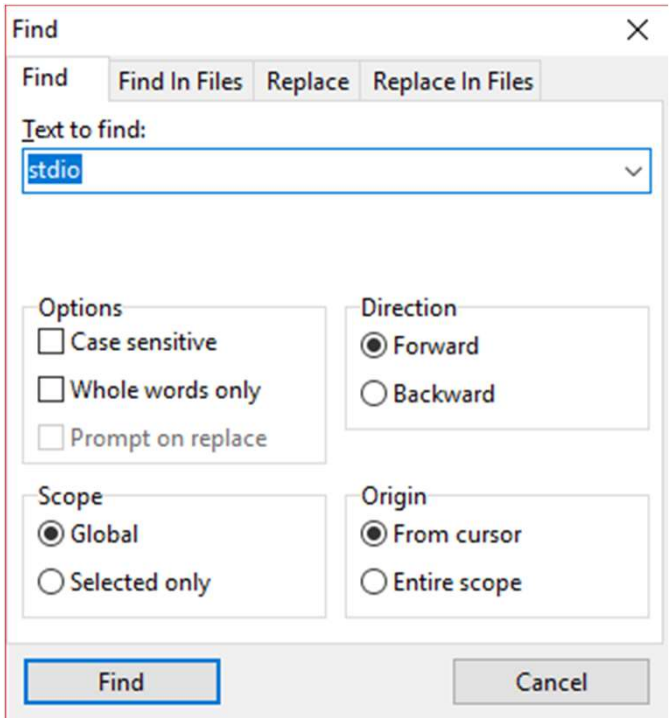


THE EDIT MENU

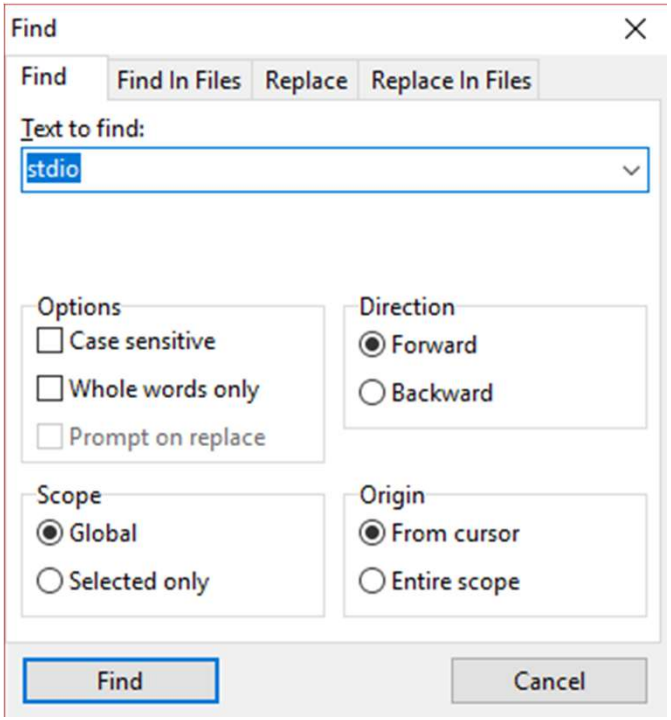


The Search Menu

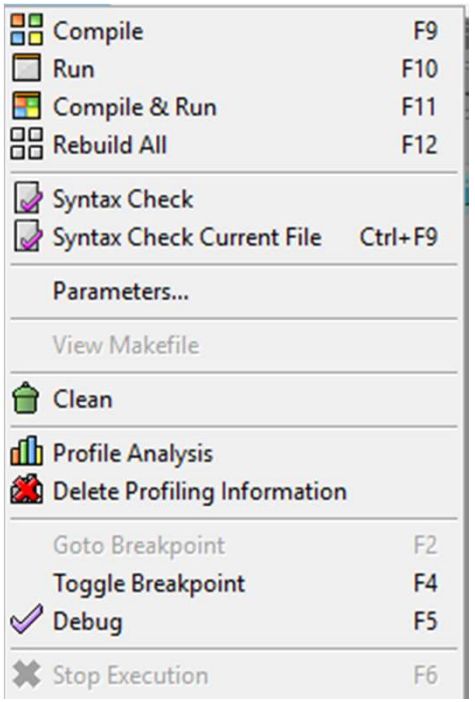




Replace text box



The Run menu


Compile F9
Run F10
Compile & Run F11
Rebuild All F12
Syntax Check
Syntax Check Current File Ctrl+F9
Parameters...
View Makefile
Clean
Profile Analysis
Delete Profiling Information
Goto Breakpoint F2
Toggle Breakpoint F4
Debug F5
Stop Execution F6

A Sample C Program

- A simple C program begins this way

```
#include <stdio.h>

int main()
{
```

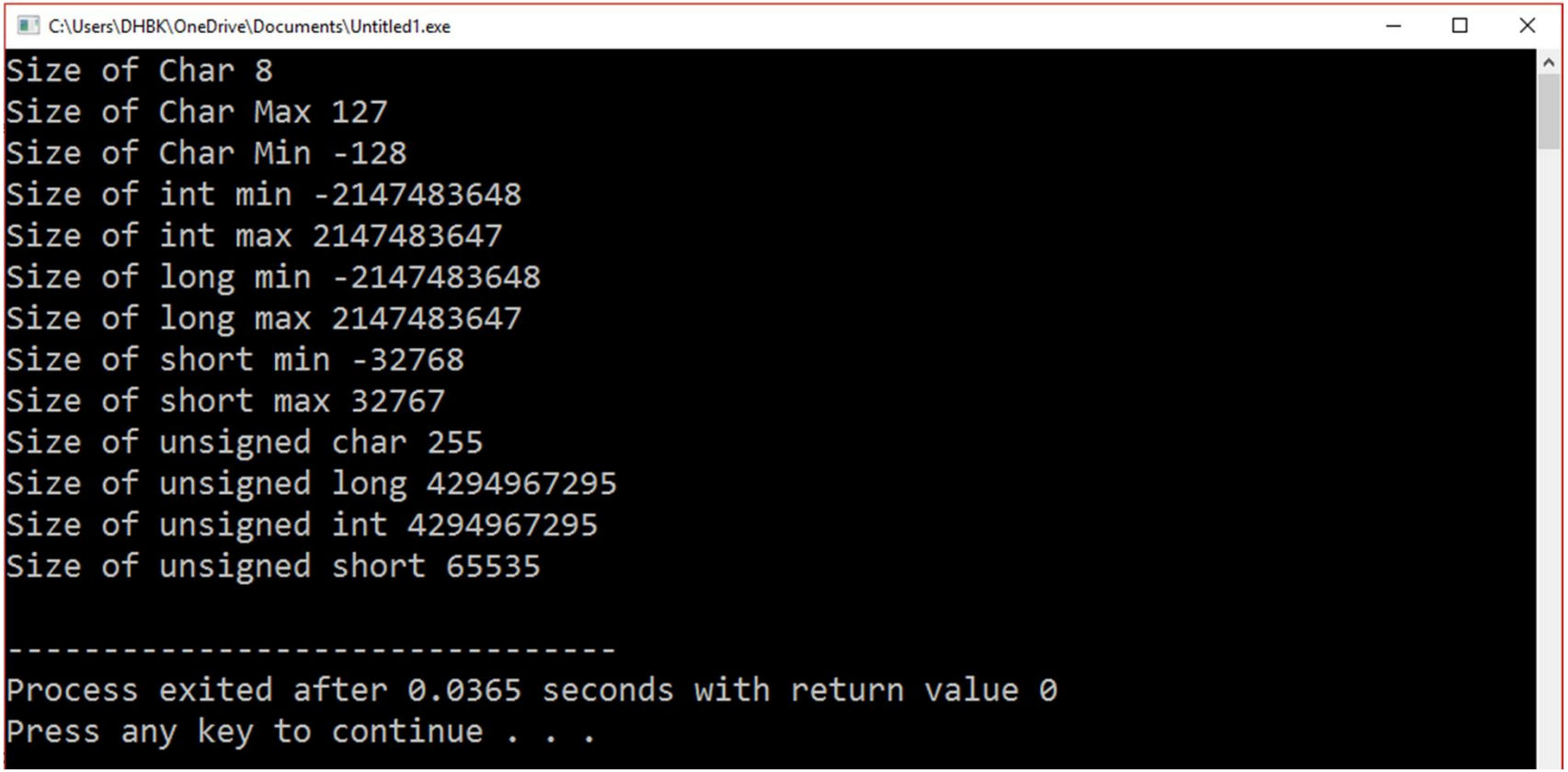
- And ends this way

```
        return 0;
}
```

A Sample C Program

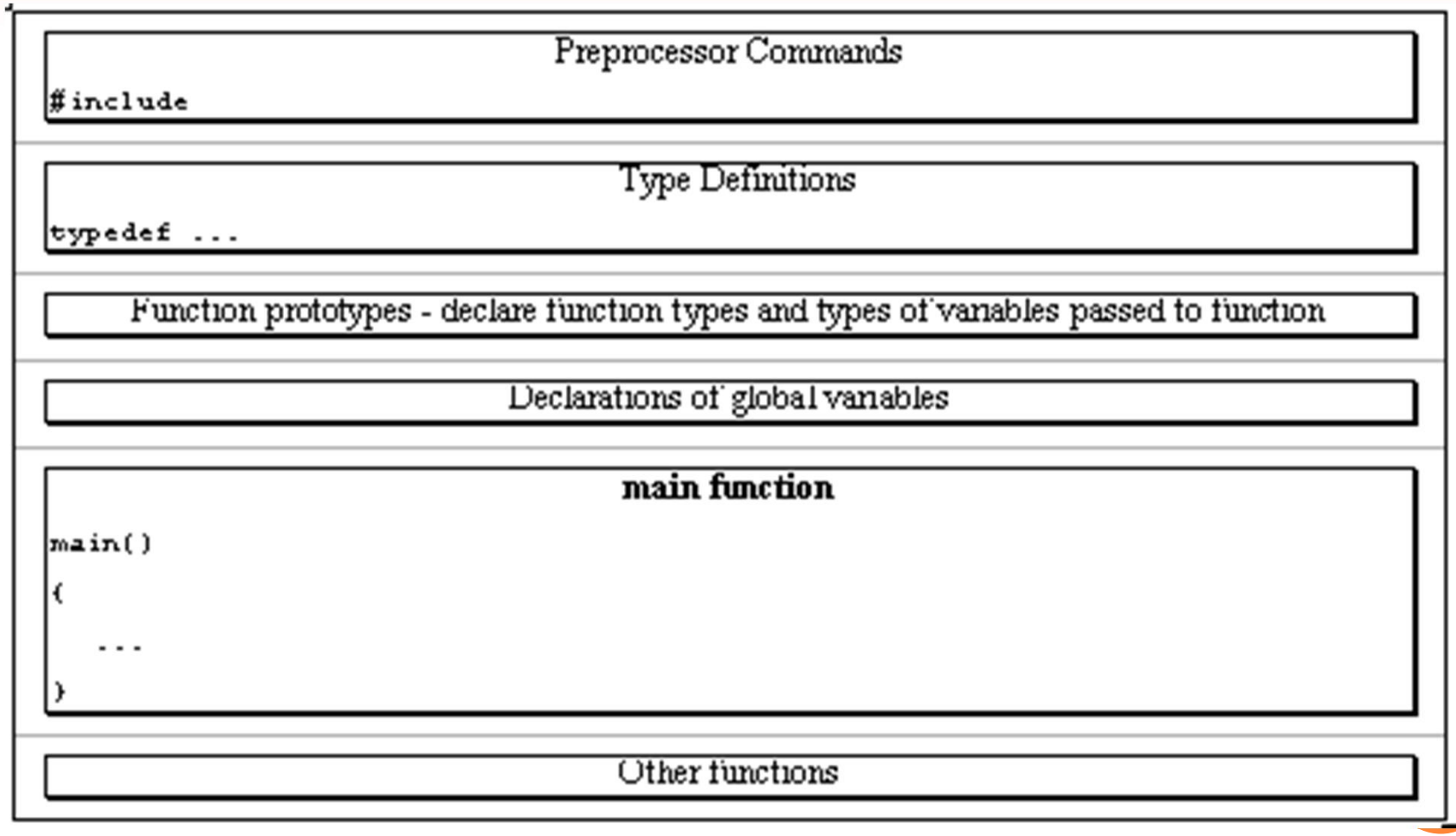
```
#include <stdio.h>
#include <limits.h>
main ()
{
  printf("Size of Char %d\n", CHAR_BIT);
  printf("Size of Char Max %d\n", CHAR_MAX);
  printf("Size of Char Min %d\n", CHAR_MIN);
  printf("Size of int min %d\n", INT_MIN);
  printf("Size of int max %d\n", INT_MAX);
  printf("Size of long min %ld\n", LONG_MIN);
  printf("Size of long max %ld\n", LONG_MAX);
  printf("Size of short min %d\n", SHRT_MIN);
  printf("Size of short max %d\n", SHRT_MAX);
  printf("Size of unsigned char %u\n", UCHAR_MAX);
  printf("Size of unsigned long %lu\n", ULONG_MAX);
  printf("Size of unsigned int %u\n", UINT_MAX);
  printf("Size of unsigned short %u\n", USHRT_MAX);

}
```

RESULT


C:\Users\DHBK\OneDrive\Documents\Untitled1.exe
Size of Char 8
Size of Char Max 127
Size of Char Min -128
Size of int min -2147483648
Size of int max 2147483647
Size of long min -2147483648
Size of long max 2147483647
Size of short min -32768
Size of short max 32767
Size of unsigned char 255
Size of unsigned long 4294967295
Size of unsigned int 4294967295
Size of unsigned short 65535

Process exited after 0.0365 seconds with return value 0
Press any key to continue . . .

C Program Structure



PRE-PROCESSOR COMMANDS

#include -- header files for library functions

Example:

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

#define -- define constants and macros

Examples:

```
#define e  2.7182818
#define pi 3.14159265359
```

VARIABLE DECLARATIONS

- Declarations tell the compiler what variable names will be used and what type of data each can handle (store).

- Example declarations:

```
int    a, b, c ;
float   r, p, q ;
double x, y, z ;
char   m, n ;
```

DATA TYPES

- Integer variables: int a, b ;
- Integer variables, like a or b, store only whole numbers like 3 or 7, not 3.33 or 7.65, and only up to certain maximum values.
- Floating point variables: float c, d ;
- Floating point variables, like c or d, store rational numbers, like 3.14159, but only a limited number of digits of precision.

Programs Have One or More Functions

- Even the main program is a function.

  The body of each user-written function is enclosed in braces, { } (or curly brackets)

- The syntax of a function is:

```
<function type> function_name  (param. list)
    {                  /* beginning of function */
    }                  /* end of function       */
```

EXECUTABLE STATEMENTS

- Simple

  Declaring variables

  ```
  int temp ;
  char a ;
  ```

  Assigning Values

  temp = 5 ; temp is assigned the value of 5

- Complex, i.e., Calling Functions

  ```
  plotxy (x, y) ;
  ```

- Calculations

  ```
  x = (5. / 2 + 6) * 7 ;
  ```

COMMENT STATEMENTS

- Formal Comments:

  /* Comment ….. */

  - Used for detailed description of functions or operations (for our benefit, not compiler's).
  - Can take multiple lines in source file.

- Informal Comments (only in C++, not C):

  // Comment ….. Ends at the end of line

  - Used for quick comments like:

  int temp; // temporary variable for storing // the input value

Hàm	Ý nghĩa
sqrt(x)	Căn bậc 2 của x //sqrt(16.0)=4.0
pow(x,y)	X mũ y (x^y)//pow(2,3)=8
exp(x)	E mũ x (e^x)//exp(1.0) = 2.718
log(x)	Logarithm tự nhiên (cơ số e) của x (*ln x*)// log
log10(x)	Logarithm cơ số 10 của x (*log x*)
sin(x)	
cos(x)	
tan(x)	
ceil(x)	Số nguyên nhỏ nhất không nhỏ hơn x ceil(2.5)=3 ceil(-2.5)=-2

Running a C Program

- C source code is written with a text editor
- The compiler on your system converts source code to object code.
- The linker combines all the object code into an executable program.

TESTING AND DEBUGGING

- Bug
  - A mistake in a program
- Debugging
  - Eliminating mistakes in programs
  - Term used when a moth caused a failed relay on the Harvard Mark 1 computer. Grace Hopper and other programmers taped the moth in logbook stating:

    "First actual case of a bug being found."

Basic Components of C Programs

- ***Symbols***
- ***Key Words***
- ***Identifiers***
- ***Data Types***
- ***Constants***
- ***Variables***
- ***Operators***
- ***Expressions***
- ***Functions***
- ***Comments***

SYMBOL

- A C program consists of the following characters:
- 26 capital letter of English alphabet : A, B, C, D, ... X, Y, Z
- 26 small letter of English alphabet : a, b, c, d, ...x, y, z
- 10 digits : 0, 1, . . . 9
- Math operators : + - * / = < >
- Other symbols

C TOKENS

- Keywords
- Identifiers
- Constants
- Strings
- Special symbols
- Operators

KEYWORDS

- *A keyword is a name which indicate a specific command. Keywords are also considered reserved words. You shouldn't use them for any other purpose in a C program.*
- The most important keywords of C are
- **asm autobreak case char const continue default do double else enum extern float for goto if int long register return short signed sizeof static struct switch type def union unsigned void volatile while**

IDENTIFIERS

Refer to a variety of things :

- Variables
- Functions
- Tag of structures
- Member of structures
- typedef names

RESTRICTIONS ON THE NAMES

- Identifiers are made up of letters and digit;
- The first character must be a letter. The underscore "_" counts as a letter
- Upper and lower case are distinct
- Only the first 31 characters are significant.
- Keywords are reserved: you can't use them as variable names.

EXAMPLES

- Valid identifiers

  i, x, b55, max_val

- Invalid identifiers

  12w (the first character is a digit)

  income tax (use invalid character " ")

  char (*char* is a keyword)

Data types

- A data type is a set of values and the operations on those values.
- Common types of data include
  - Primitive types
  - Structured types

CONSTANTS

- A constant is a specific quantity that does not or cannot change or vary.
- C supports three types of constants : numeric, character, string.
  - Numeric constants just the written version of numbers.
    Examples 1, 0, 56.78, 12.3e-4.
  - Octal constants are written with a leading zero : -0.15
  - Hexadecimal constants are written with a leading 0x : 0x1ae
  - Long constants are written with a trailing L : 890L or 890l

CHARACTER AND STRING CONSTANTS

- **Character constants are usually the character enclosed in single quotes; 'a', 'b', 'c'.**
- **Some characters use a 2 character sequence (escape sequence).**
  - **'\n' newline**
  - **'\t' horizontal tab**
  - **'\v'vertical tab**
  - **'\b'backspace**
  - **'\r'carriage return**
  - **'\\'backslash**
  - **'\"single quote**
  - **'\'"double quotes**
  - **'\0'null**
- **Character constants participate in numeric operations just as any other integers**
- **A string constant is a sequence of characters surrounded by double quotes e.g. "Brian and Dennis".**
- **A character is a different type to a single character string**

Operators

- Arithmetical operators
- Relational operators
- Boolean operators
- Assignment operators
- Bitwise operators

EXPRESSIONS

- Arithmetic expression
- Conditional expression
- Assignment expression

SUBPROGRAMS

- A collection of instructions forming a program unit
  - written independently of the main program
  - associated with the main program through a transfer/return process.
- Control is passed to the subprogram at the time its services are required,
- Control is returned to the main program after the subprogram has finished.

SUBPROGRAMS: ADVANTAGES

- Reducing the duplication of code in a program
- Enabling reuse of code across multiple programs,
- Decomposing complex problems into simpler pieces
- Improving readability of a program,
- Hiding or regulating part of the program

SUBPROGRAMS

- In many programming language, subprograms are divided into
  - Functions: generate return values and appear in expressions
  - Procedures: generate no return values and appear in statements
- An item of data in either the main program or the subprogram is not automatically accessible from within the other.
- Any transfer of data is specified by the programmer
- This is done by parameters

FUNCTIONS

- C only accept one kind of subprogram: function.
- Information is returned from a function to the main program in the form of the "value of the function"
- A function can be called by other functions or by itself (recursion)

TURBO C STANDARD LIBRARIES

- ***stdio.h*** Provides functions for performing input and output.
- ***stdlib.h*** Defines several general operation functions and macros.
- ***conio.h*** Declares several useful library functions for performing "console input and output" from a program.
- ***math.h*** Defines several mathematic functions.
- ***string.h*** Provides many functions useful for manipulating strings (character arrays).
- ***graphics.h*** Includes graphics functions