Chapter - 3
C Programming Tokens, Data Types

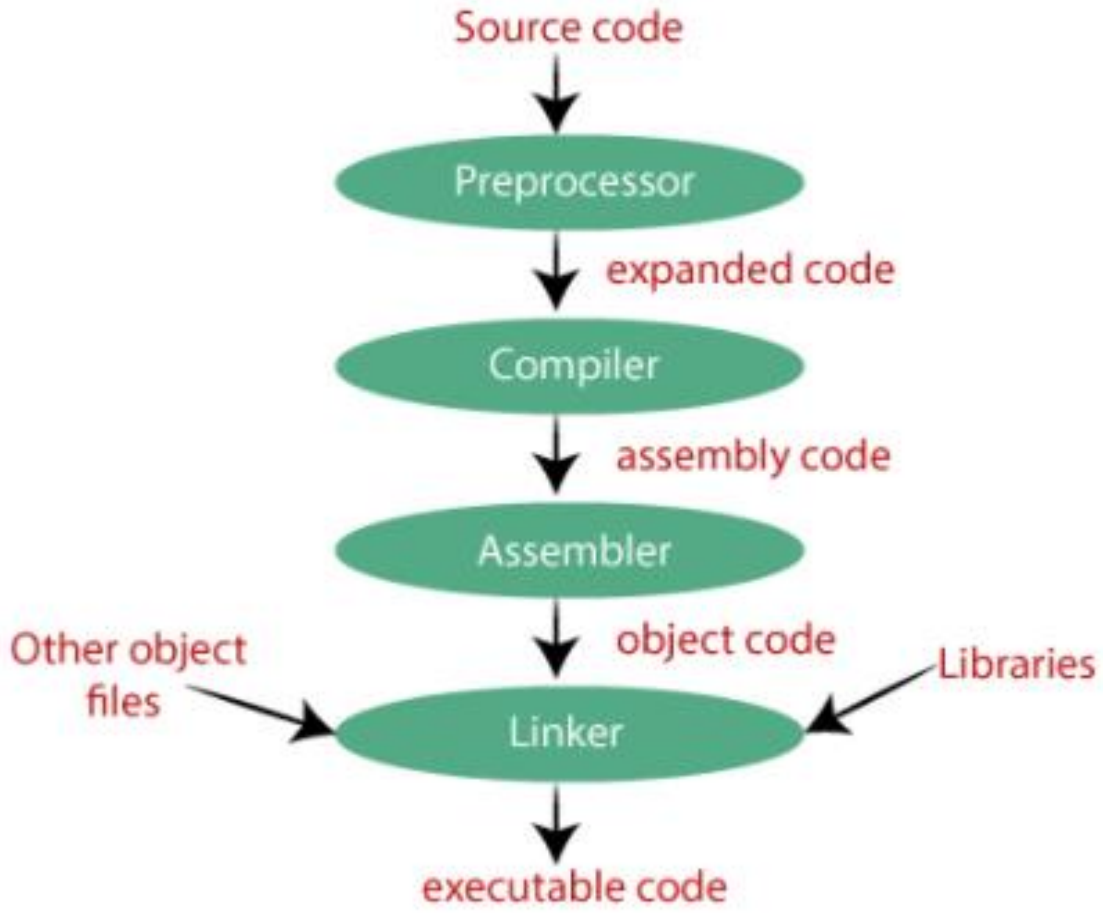
Course Code: CIS 115 & 115 L

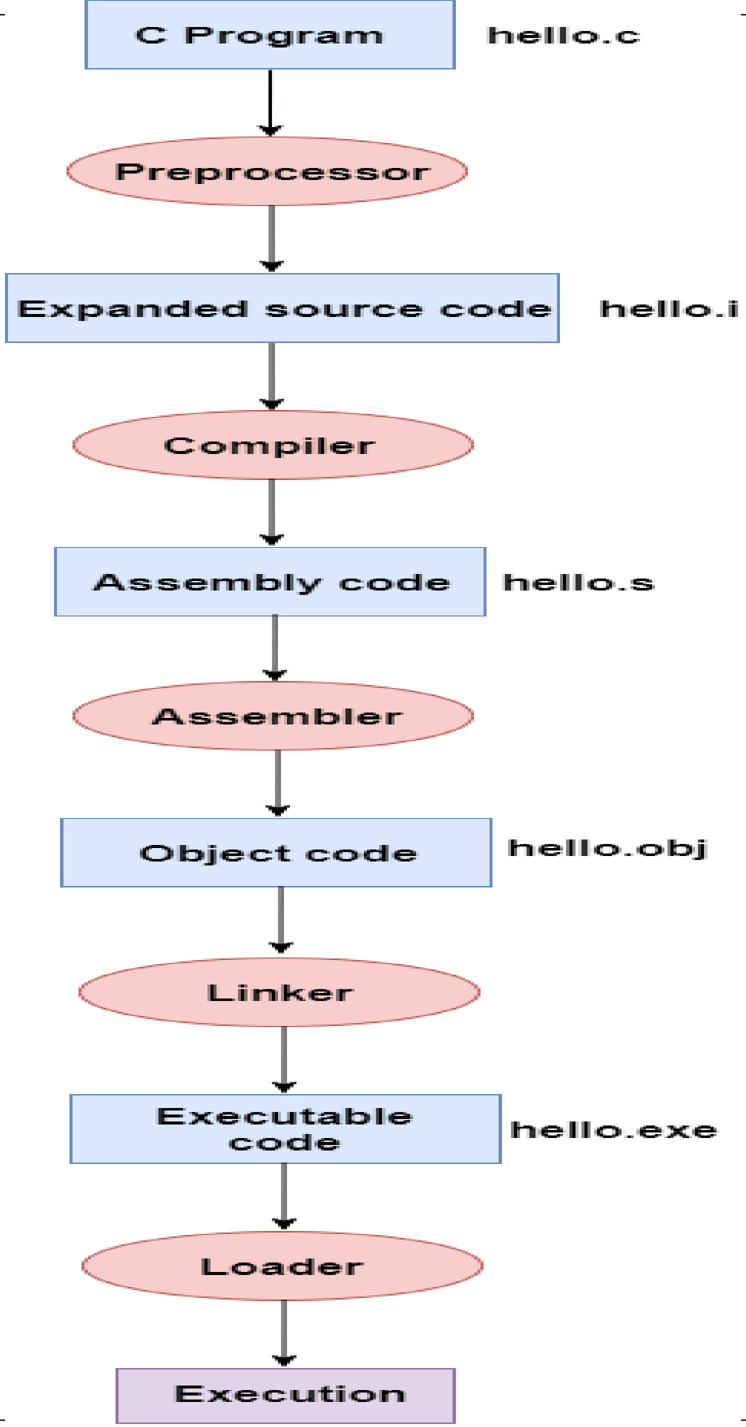
Course Title: Structured Programming

Course Teacher: Md. Faruk Hosen

Prepared by: ABK Bhuiyan (Jehad)

C Compilation Process





C Program

hello.c

Preprocessor

Expanded source code

hello.i

Compiler

Assembly code

hello.s

Assembler

Object code

hello.obj

Linker

Executable code

hello.exe

Loader

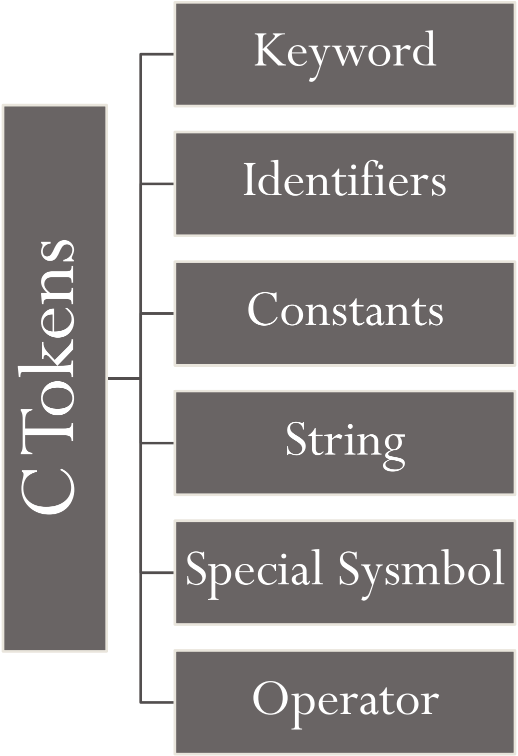
Execution

C TOKENS

- For understanding C tokens first we have to understand the words or sequence of characters that are used or supported by C. In C programming different character set are used which are the following type:
- Letter: All alphabets. Uppercase A - Z and lowercase a – z.
- Digit: All decimal digits 0…….9
- Special Characters: Comma (,). Semicolon (;), Colon (:), Question mark (?), dollar sign ($), back slash (\), slash (/), exclamatory sign (!), vertical bar (|), Percentage sign (%).
- White Spaces: Blank spaces, New Line, Carriage Return, Form Feed, Horizontal tab.

Types of Tokens

In a text each of the individual words and punctuations marks are called tokens. In C programming each of the smallest individual unit is called token. C tokens are classified in six types as follows:



Keyword

In programming C there is a list of words that have specific meaning and task and the meaning of these words cannot be changed called keyword. Keywords are the building blocks of program instructions. All keywords in C are written in lower case. **Some of the C keywords** are given below:

int	**float**	**double**	**char**
while	for	auto	extern
case	const	default	goto
if	do	static	switch

Notion of keywords

- Keywords are certain reserved words, that have standard predefined meanings in C.
- All keywords are in lowercase.
- Some keywords in C:

auto extern sizeof break

static case for

struct goto switch

const if typedef enum

signed default int union

long continue

unsigned do register void

double return volatile else

short while float char

Identifiers & Variables

- Identifier : A name has to be devised for program elements such as variables or functions.
- Variable: Variables are memory regions you can use to hold data while your program is running.
- Thus variable has an unique address in memory region.
- For your convenience, you give them names (instead of having to know the address)
- Because different types of data are different sizes, the computer needs to know what type each variable is – so it can reserve the memory you need

Rules for Identifiers

Identifiers are the name of the variables and sub programs or functions. These are user-defined names means written by users, but there are some rules that must be maintained when writing identifiers. Rules are as follows:

- Contains only letters, digits and under score characters,

  example amount, hit_count
- **Must begin with** either a letter of the alphabet or an underscore character.
- Can not be same as the keywords, **example it can not be** void, int.
- Uppercase letters are different than lowercase, example **amount, Amount and AMOUNT** all three are different identifiers.
- **Maximum length** can be **31** characters.
- Should not be the same as one already defined in the library, example it can not be printf/scanf.
- **No special characters are permitted**. e.g. blank space,period, semicolon, comma etc.

Constants

In programming C constants refers to the fixed values that can't be changed during the execution of a program. Example: Value of pi 3.1416. The C constants are classified as:

INTEGER CONSTANTS

- Integer constants refer to the numeric constants values without having fractional or exponential values. Integer constants in C can be divided into three types:
- Decimal: Consist of digits 0 to 9 and can be signed + or -. Example: 100, -20, +50 etc.
- Octal: This constants type holds values 0 to 7. Example: 007, 702 etc.
- Hexadecimal: These types of integer constants are rarely used. Values contains 0 to 9 and A to F. Example: 0x7F, 0x6A etc.

Constants

STRING CONSTANTS

- String Constants is sequence of characters enclosed in double quotation. Example: “Hello”, “1971” etc.

ESCAPE SEQUENCES

- In c there are some back slash character constants in output functions those are known as Escape sequences. Example: for output in new line ‘\n’ is added. The lists of characters are:
- are
- those are

Constants (ESCAPE SEQUENCE)

Constant	Meaning
‘\a’	Audible Alert
‘\b’	Back Space
‘\f’	Form Feed
‘\n’	New Line
‘\r’	Carriage Return
‘\t’	Horizontal Tab
‘\v’	Vertical Tab
‘\”	Single Quote
‘\”’	Double Quote
‘\?’	Question Mark
‘\\’	Back slash
‘\0’	Null

Data Types

- Every program specifies a set of operations to be done on some data in a particular sequence.
- However, the data can be of many types such as a number, real, character, string etc.
- C supports many data types out of which some basic types are:

  int, float , double, char & void.

- Each of them is used in programming C for declaring different types of variable for holding different values. As like for storing price of a fish we have a variable price. But what type of data it will hold we have to define. As price of a fish can be fractional so its data type will be floating. Now we will learn brief about all the types.

DATA TYPES

- **INTEGER TYPE**

Integer types refers to the numeric values without fractional or exponential values In C integers are declared with the keyword **“int”.** Size or range of integer data types depends on the computer, it differs depending on word length.

There are some extension of integer for controlling the size and storage space. They are short int, long int, signed int, unsigned int.The size of them will be discussed later.

DATA TYPES

- **FLOATING**

Floating type refers to the numeric values having fractional and exponential numbers with six digit of precision. The floating types are defined with keyword **“float”** in C.

When we need more precession for perfect value than we have use double data types which is an extension of floating type defined with keyword **“double”.**

Double data types have 14 digit precision. For larger value there is another type called long double which defining keyword is **“long double”**with 19 digit of precision.

DATA TYPES

- **CHARACTER**

For declaring single character data character data type is used. Keyword for defining character data type is **“char”.** Character can be signed or unsigned. Unsigned character ranges from 0 to 255 and signed character ranges from -128 to 127.

- **VOID TYPES**

Void types has no values. For defining different functions we need to use this type. If any sub program or function does not return any value then its type is defined as void. The keyword for defining void is **“void”**.

DATA TYPES

The **size of different data types** are given below:

Data Type	Size	Range
char	8 bits	0 – 255 or -128 – 127
unsigned char	8 bits	0 – 255
signed char	8 bits	-128 – 127
int	16 bits	-32768 – 32767
signed int	16 bits	-32768 – 32767
unsigned int	16 bits	0 – 65535
short int	8 bits	0 – 255
long int	32 bits	-2147483648 - 2147483647
float	32 bits	1.2E-38 to 3.4E+38
double	64 bits	2.3E-308 to 1.7E+308
long double	80 bits	3.4E-4932 to 1.1E+4932

VARIABLES

Variable is a container that holds value of a specific type. Variable store values. Actually, the name of the variable is a reference of a memory location for storing data.

The value of a variable can be changed during the execution of a program. Variable name is defined by programmer. Example: Salary, interest etc.

- To Declare a variable means to **reserve memory space** for it.

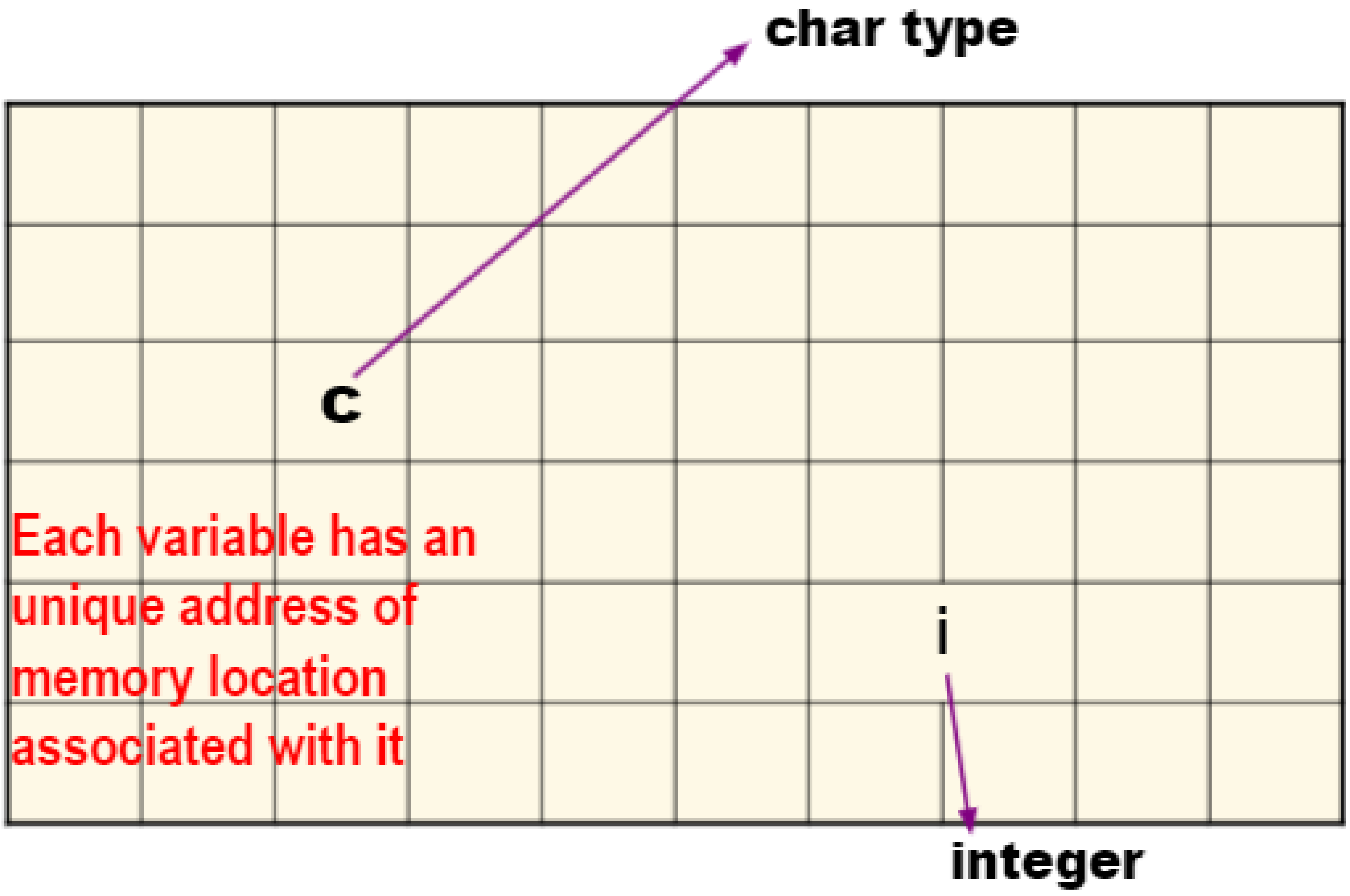
DECLARING VARIABLE

- General format for declaring a variable is: **data-type variable-name;**
- Example: Declaring a integer data named amount: int amount;
- For declaring more than one variable: **data-type variable1,variable2, … ,variableN;**
- Examples: int x,y,z;

  double area;

  float perimeter;

Memory view



Each variable has an unique address of memory location associated with it

Important about Variable

- Always remember in C , a variable must always be declared before it used.
- Declaration must specify the data type of the value of the variable.
- Name of the variable should match with its functionality /purpose.

Example int count ; for counting the iterations.

float per_centage ; for percentage of student

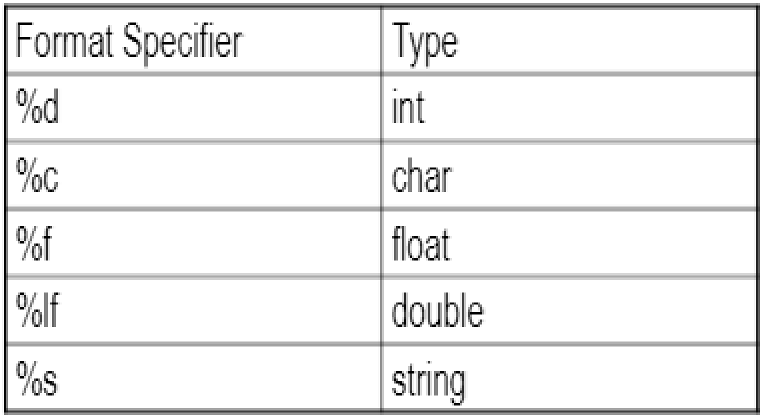
Initialization

- Initializing a variable involves assigning(putting in) a value for the first time. This is done by using the ASSIGNMENT OPERATOR, denoted by the equals sign,=.
- Declaration and initializing can be done on separate lines, or on one line.
- char c='x'; or

```
char c;
c='x'
```

Format specifiers

- There are several format specifiers-The one you use should depend on the type of the variable you wish to print out. The common ones are as follows:



Format Specifier	Type
%d	int
%c	char
%f	float
%lf	double
%s	string

- To display a number in scientific notation, use %e.
- To display a percentage sign, use %%

Using printf()

1. It is used to print message or result on the output screen. It is define in **stdio.h** header file.

2. Returns the number of characters it outputs on the screen.

- Example:

  printf(“Enter the number whose multiplication table you want to generate”);

  printf(“ Balance in your account is %d”, bal); /* where bal is int type*/

  printf(“Balance =%d, Total tax is %f ” , bal, tax); /* where tax is float type */

Using scanf()

- scanf( ) : It is used to input from the user numerical values, characters or strings.It is defined in stdio.h
- The function returns the number of data items that have been entered successfully.
- **Example:**

```
int num1,num2,num3;
 char var;
printf(“ enter the numbers “);
scanf(“%d%d%d”, &num1,&num2,&num3);
printf(“enter a character”);
scanf(“%c”, &var);
```