

Course : CSE 105

Structured Computer Programming

Khaled Hussain

BSc Hons (UK), MSc (UK)

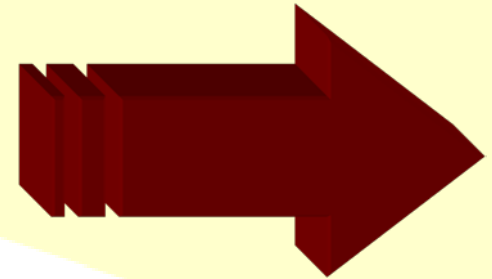
Associate Professor & Dean

School of Science & Engineering

Sylhet International University

C

Fundamentals



Recommended Books

- E. Balagurusami - *Programming in ANSI C 5th Ed.*
- Byron Gotteried - Schaume's Outlines – *Programming with C 3rd Ed.*
- Herbert Schildt - *Teach Yourself C*
- Any other book you like 😊



Use of Internet is highly encouraged !



**You must have a pen drive of your own.
This is MANDATORY for this course !!!**

Before we continue...

- You should be familiar with the operating system on your PC !
 - *Different types of extensions associated with files (.doc, .jpg, .exe, .c etc.)*
 - *Organization of files and folders in the hard drive*
 - *Copy / paste files and folders to different destinations*
 - *Find specific files in the hard drive etc.*
- You should be able to install simple software by yourself

A Simple C Program

```
/* This program displays a message on the screen */  
  
#include <stdio.h>  
  
int main (void)  
{  
    printf("Welcome to CSE 105");  
    return 0;  
}
```

Welcome to CSE 103

'C' Character Set

These are the characters that C recognizes.

- **Letters** (upper case and lower case)

```
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w x y z
```

- **Digits**

```
0 1 2 3 4 5 6 7 8 9
```

- **Special Characters**

```
" ( ) * + - / : = ! & $ % ? , .
^ # @ ~ ` { } [ ] \ | (blank space)
```

Identifiers

Identifiers allow us to name data and other objects in the program. Each identified object in the computer is stored at a unique address.

- Identifiers consists of letters and digits except that the first character MUST be a letter.
- Both uppercase (CAPITAL) and lowercase (small) are acceptable though lowercase is desirable in most cases.
- The underscore (_) can also be included which can be placed in the middle of an identifier or at the beginning of an identifier.

C is case sensitive !

These are valid identifiers –

x y2 name address1 total _sum

income_tax STUDENT

The following are not valid identifiers -

2a

The first character is not a letter

roll no

Illegal character (blank space)

total-Marks

Illegal character (-)

Keywords

- Keywords are reserved words that have standard, predefined meaning in C and are used for specific purposes.
- Keywords are all lowercase!

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

Variables & Data Types

- A variable is an identifier that is used to represent a single data item. It is a named area in the computer memory.
- Variables are placeholders for values; just like in algebra

$$\text{Total} = \text{Sessional} + \text{Midterm} + \text{Final}$$

- *They can be used to store data to be used elsewhere in the program.*

- Variables have specific *data types*

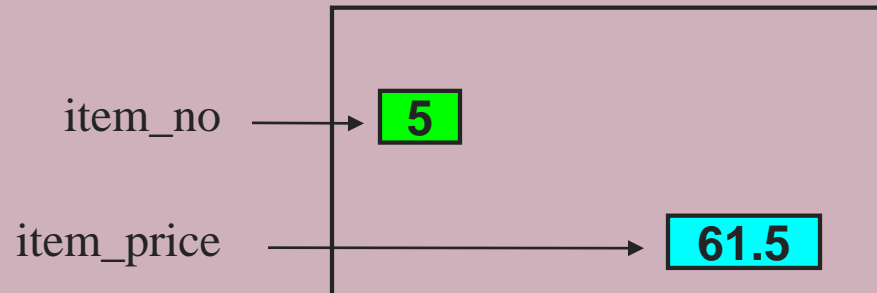
int - Integer numbers (24, 512, -98)

char - ASCII character (a, b, x, y)

float - Decimal fractional numbers (2.14, 0.5, -100.6)

Variables in Memory

```
int item_no = 5;  
float item_price = 61.5;
```



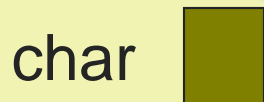
- Variables have specific *data types*

int - Integer numbers - 2 to 4 bytes (24, 512, -98)

char - ASCII character – 1 byte (a, b, x, y)

float - Decimal fractional numbers – 4 bytes (2.14, 0.5, -100.6)

double - Decimal fractional numbers – 8 bytes



- **To declare a variable**

Format :

dataType variable1, variable2, variable3;

Example

int total, counter ;

You can initialize values at the same time

Example

int total=0, counter=0;

Constants

- A constant is an identifier like a variable except that the value of a constant cannot change.
- Four basic types of constants -

Integer constants

(103, 78, -5672)

Floating-point constants

(2.7, 124.5, -345)

Character constants

('A', 'q', '7', '?')

String constants

("Hi", "Hello world")



,



”

Categories of Constants

- **Defined Constants** – Using a preprocessor command *define*.

```
#define PI .25
```

- **Memory / Named Constants** – Using a type qualifier.

```
const float pi = 3.14159
```

- **Literal Constants** – An unnamed constant used to specify data.

```
a +5  
3.1416  
'A'
```

The three ways of coding constants

```
#include <stdio.h>

#define PI 3.14159

int main(void)
{
    const double pi = 3.14159;
    printf("Literal Constant PI = %f\n", 3.14159);
    printf("Defined Constant PI = %f\n", PI);
    printf("Memory Constant PI = %f\n", pi);
    return 0;
}
```

Statements

A statement causes an action to be performed by the program. In other words, a statement instructs the computer to do something.

```
printf("Enter product code");  
total_amount = rate * quantity;  
circumference = 2 * pi * radius;
```


Main three types of statements -

- ***Expression Statement***

Consists of an expression followed by a semicolon.

```
rate = 95;  
printf("Enter product code");
```

- ***Compound Statement***

Consists of several individual statements enclosed within a pair of braces { }. A compound statement **does not** end with a semicolon.

```
{  
    pi = 3.14159;  
    circumference = 2 * pi * radius;  
    area = pi * radius * radius;  
}
```