

MNNIT COMPUTER CODING CLUB

CLASS-1

BASICS OF C



INSTALLATION

- For Windows (additional step)
 - https://linuxhint.com/install_ubuntu_windows_10_wsl/
- Installation of gcc compiler in Linux
 - `sudo apt update`
 - `sudo apt install build-essential`
 - `sudo apt-get install gedit`
- Check if gcc is working
 - `gcc --version`

PROGRAMMING LANGUAGE: C

- C is developed by Dennis Ritchie in 1972.
- C is a general purpose, procedural computer programming language.
- C is a high-level language. A high-level language (HLL) is a programming language such as C, FORTRAN, or Pascal that enables a programmer to write programs that are more or less independent of a particular type of computer. Such languages are considered high-level because they are closer to human languages and further from machine languages.
- Roadmap to study languages
 - Step 1: Study a simple procedural language first. Study C language properly which will be helpful to you throughout your academics as well.
 - Step 2: Study object-oriented languages such as Java, Python, C++ etc.

HELLO WORLD

- Step 1: Type this code in a code editor. You can use Sublime Text , notepad or any other editor of your choice.
- Step 2: Create a folder named **practice** in desktop.
- Step 3: Save the file with name **helloworld.c** in this folder.
- Step 4: Type command
 - gcc helloworld.c
 - Check for any error
- Step 5: To run the program type command
 - ./a.out

```
1  #include<stdio.h>
2
3  int main()
4  {
5      printf("Hello World");
6      return 0;
7  }
8
```

Output : **Hello World**

Points To Notice about structure of a C program:

1. Preprocessor directives (#)
2. Header File
3. Main Function
4. return at the end of program.

HELLO WORLD

<code>#include<stdio.h></code>	Loads the required files to write code
<code>int main()</code>	Contains your code
<code>printf()</code>	Gives output
<code>Return 0</code>	Done with your code

KEYWORDS

- reserved for doing specific tasks
- standard, predefined meaning in C
- always written in lowercase
- only 32 keywords available
- e.g. int, return, short, void

DATA TYPES

- int, char, float and double
- 'char' is used to store any single character
- 'int' is used to store integer value
- 'float' and 'double' is used for storing decimal values
- Size qualifiers - short, long
- Sign qualifiers - signed, unsigned

- For e.g.) `int x = 5;`

Basic data types	Data types with type qualifiers	Size(bytes)	Range
char	char or signed char	1	-128 to 127
	unsigned char	1	0 to 255
int	int or signed int	2	-32768 to 32767
	unsigned int	2	0 to 65535
	short int or signed short int	1	-128 to 127
	unsigned short int	1	0 to 255
	long int or signed long int	4	-2147483648 to 2147483647
	unsigned long int	4	0 to 4294967295
float	float	4	3.4E-38 to 3.4E+38
double	double	8	1.7E-308 to 1.7E+308
	long double	10	3.4E-4932 to 1.1E+4932

VARIABLES

- name that can be used to store values
- can take different values but one at a time
- A data type is associated with each variable
- data type of the variable decides what values it can take

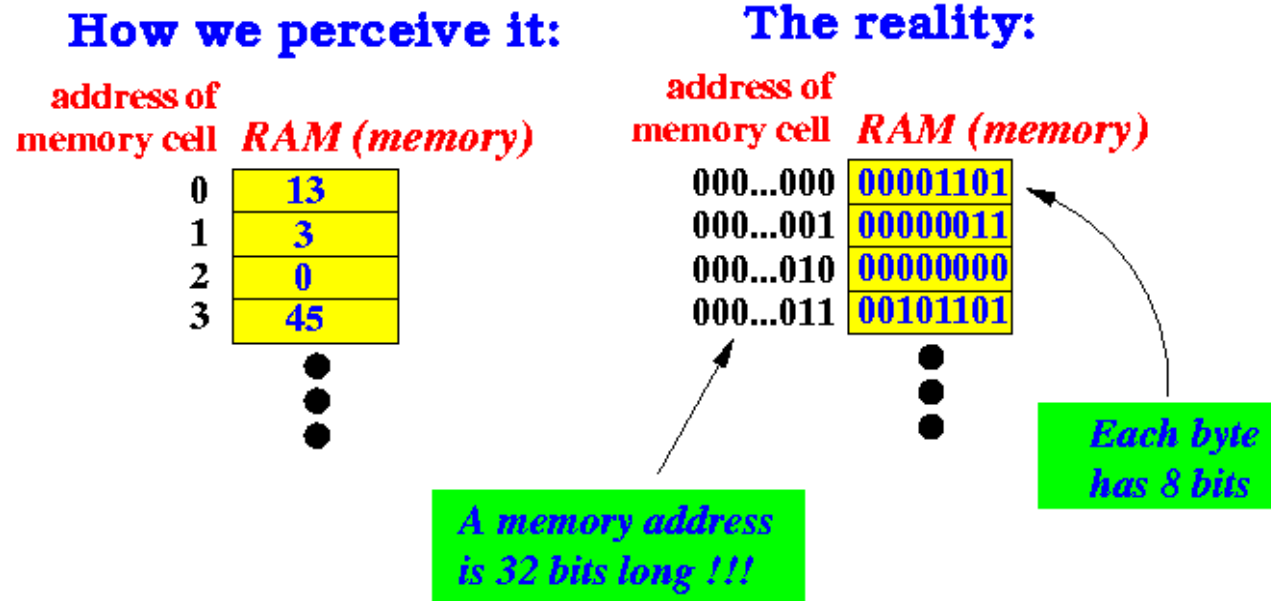
IDENTIFIERS

Identifiers are the names given to variables and other things. Rules for an Identifier-

- only have alphanumeric characters(a-z , A-Z , 0-9) and underscore(_).
- The first character can only contain alphabet(a-z , A-Z) or underscore(_).
- Identifiers are also case sensitive in C. For example **name** and **Name** are two different identifiers in C.
- Keywords are not allowed to be used as Identifiers.
- No special characters, such as semicolon, period, whitespaces, slash or comma are permitted.

LANGUAGE OF COMPUTERS: BINARY

- Electronic devices can understand only 2 states - **ON** or **OFF** || **0** or **1**
- Data stored in the memory of computer is in the form of 1 or 0.
- Bit : The bit is a basic unit of information in computing. A bit stores just a 0 or 1.
- Byte: combination of 8 bits.



INPUT-OUTPUT IN C

- **Input** - data given by user to program in execution
 - Can be given in the form of file or from command line.
 - Set of built-in functions to read input.
 - ***scanf("%X", &variableOfXType);***
 - **%X** is format specifier , **&** is address operator

```
1  #include<stdio.h>
2  int main(){
3      int a;
4      scanf("%d",&a);
5      return 0;
6  }
7
```

INPUT-OUTPUT IN C

- **Output** – It means to display some data on screen or file.
 - Can be given in the form of file or from command line.
 - Set of built-in functions to display output.
 - ***printf("%X", variableOfXType);***
 - **%X** is format specifier , **&** is address operator

```
1  #include<stdio.h>
2  int main(){
3      int a;
4      scanf("%d",&a);
5      printf("Output is : %d",a);
6      return 0;
7  }
8
```

OPERATORS

- Operator is a symbol to perform specific mathematical or logical functions.
- For eg '+' operator adds two operands.
- Types of operator :
 - Arithmetic Operators (+, -, /, *)
 - Relational Operators (<, >, <=, >=, ==)
 - Logical Operators (&&, ||)
 - Bitwise Operators (^, |, &, ~)
 - Assignment Operators (=)
 - Misc Operators
- Operators have a specific precedence and associativity which will be discussed later.

STATEMENTS

- In C program , instructions are written in the form of statements.
- Examples of statements :
 - `int x = 5;`
 - `func(a,b);`
 - `x= y- z;`
- Statements insides {} curly braces are known as compound statement.

COMMENTS

- In C program , comments are used to increase readability.
- Examples of comments:
 - `// this is a single line comment`
 - `/* This is also a comment */`
- It is not allowed to nest comments in C.

GET BOOKS FROM

- http://paperfactorymnnit.pythonanywhere.com/paper_mnnit/comp_books/