



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 3

Name..... Faculty No..... Class..... Batch.....

Steps in learning C:

alphabets	constants		
Digits	Variables	Instructions	Programs
special-sy-	Keywords		
tools			

A computer program is just a collection of the instructions necessary to solve a specific problem. The basic operations of a computer system form what is known as the computer's instruction set. And the approach or method that is used to solve the problem is known as an algorithm.

So far as programming language concern these are of two types:

1. Low level languages
2. High level language.

1. Low level language:-

Low language are machine level and assembly level language. In machine level language computer only understand digital numbers i.e in form of 0 and 1.



M.B.M. UNIVERSITY JODHPUR

Date: _____ Experiment No. _____ Page: 5
Name: _____ Faculty No. _____ Class: _____

is software which is used to translate high level language as well as low level language to machine level language.

Three types of translation are there:—

Compiler	Interpreter	Assembler
----------	-------------	-----------

Compiler and interpreter are used to convert the high level language into machine level language. The program written in high level language is known as source program and the corresponding machine level language program is called as object program. Both compiler and interpreter perform the same task but there.

Interpreter perform the same task but there. Interpreting is different. Compiler read the program at a time and searches the error and tells them if the program is error free then it is converted into object program. When program size is large then compiler is preferred. Whereas interpreter read only one line of the source code and convert it to object code. If it is not error, statement by a statement



M.B.M. UNIVERSITY
JODHPUR

Date _____ Experiment No. _____
Name _____ Faculty No. _____ Class _____ p No. 6

and hence it takes more time

→ Integrated Development Environment (IDE)
The process of editing, compiling, running and debugging program is often managed by a single integrated application known as an Integrated Development Environment, or IDE for short.
An IDE is a windows-based program that allows us to easily manage large software programs, edit files in windows, and compile, link, run, and debug programs.

_____ x _____



M.B.M. UNIVERSITY JODHPUR

Name _____ Roll No. _____ Page No. _____
Department No. _____ Faculty No. _____ Date _____

As instructions given to the computer are in the form of binary digits, which is difficult to implement. Instructions in binary code. This type of program is not portable, difficult to write, and also error prone.

The assembly language is an other level modified version of machine level language. Where instructions are given in english like word as ADD, SUB, MOV etc. It is easy to write and understand but not understood by the machine. So the translator used here is an assembler to translate into machine level. Although language is bit easier, programmer has to know low level details related to low level language. In the assembly level language the data are stored in the computer registers, which varies for different computer. Hence it is not portable.

High level language :-

These languages are machine independent means it is portable. The language in this category is Pascal, C, C++, Fortran etc. High level languages are not understood by the machine, so it need to translate by the translator into machine level. A translator



M.B.M. UNIVERSITY JODHPUR

Date _____ Experiment No _____ P No 7
Name _____ Faculty No _____ Class _____ Batch _____

Structure of C language program.

- 1) comment line
- 2) Preprocessor directive
- 3) Global variable declaration
- 4) main function ()

{

Local Variables;

statements;

}

user defined function

}

}

comment line.

It indicates the purpose of the program. It is represented as `/* ... */`

comment line is used for increasing the readability of the program. It is useful in explaining the program and generally used for documentation. It is enclosed with in the delimiters. Comment line can be anywhere in the program except inside string constant & character constant.

—



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 8

Name..... Faculty No..... Class..... Batch.....

Preprocessor Directive :-

#include <stdio.h> tells the compiler to include information about the standard input/output library. It is also used in symbolic constant such as #define PI 3.14 (Value).

The stdio.h (standard input output header file) contains definitions & declarations of system defined functions such as printf(), scanf(), pow() etc. Generally printf() function used to display and scanf() function used to read value.

Global Declaration :-

This is the section where variable are declared globally so that it can be access by all the functions used in the program. And it is generally declared outside the functions.

main()

It is the user defined function and every function has one main() function from where actually program is started and it is enclosed with in the pair of curly braces.



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 9

Name..... Faculty No..... Class..... Batch.....

The main() function can be anywhere in the program but in general practice it is placed in the first position.

Syntax:

main()

{

.....

.....

.....

}

The main() function return value when it declared by data type as int main()

{

return 0

}

The main function does not return any value when void (means null / empty) as void main()

{

printf("C language");

}

Output: C language



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 10

Name..... Faculty No..... Class..... Batch.....

The program execution start with opening brace and end with closing brace.

And in between the two braces declaration part as well as executable part is mentioned. And at the end of each line, the semi-colon is given which indicates statements termination.

/* First C program with return statement */.

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

```
int main (void) {
```

```
{
```

```
printf ("Hello World !! \n"); return 0;
```

```
}
```

Output: ~~Welcome to C programming language.~~



M.B.M. UNIVERSITY JODHPUR

Aashita Photography - 9828609100

Date _____ Experiment No. 2 Page 17
Name _____ Faculty No. _____ Class _____ Batch _____

Programme Name: C++ Write a programme to add two integers.

Algorithm:-
Step 1 $\rightarrow$ Start
Step 2 $\rightarrow$ Input two no. say a & b
Step 3 $\rightarrow$ $sum = a + b$
Step 4 $\rightarrow$ Display sum of two number
Step 5 $\rightarrow$ Stop

Source code:-
#include <stdio.h>
#include <conio.h>
void main()
{
int a, b;

~~clear~~
Print ("Enter the value of a and b \n");
scanf ("%d %d", &a, &b);
Print ("sum = %d", a + b);
~~getch()~~
}

Output:- Enter the value of a and b
16 4
SUM = 20



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 4

Name..... Faculty No..... Class..... Batch.....

Steps for Compiling and executing the program

A compiler is a software program that analyses a program developed in a particular computer language and then translates it into a form that is suitable for execution on a particular computer system. Figure below shows the steps that are

involved in entering, compiling and executing a computer program developed in C programming language and the typical UNIX commands that would be entered from the command line.

Step 1:- The program that is to be compiled is first typed into a file on the computer system. There are various conventions that are used for naming files, typically be any name provided the last two characters are ".c" or file with extension :c. So, the file name prog1.c might be a valid filename for a C program. A text editor is usually used to enter the C program into a file. For example, vi is a popular text editor used on UNIX systems. The program that is entered into the file is known as the source program because



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 12

Name..... Faculty No..... Class..... Batch.....

It represents the original form of the program expressed in the C language.

Step 2:— After the Source program has been entered into a file, then proceed to have it compiled. The compilation process is initiated by typing a special command on the system. When this command is entered, the name of the file that contains the source program must also be specified. For example, under UNIX, the command to initiate program compilation is called `cc`. If we are using the popular GNU C compiler, the command we use is `gcc`.

Typing the line

`gcc prog1.c or cc prog1.c`

In the first step of the compilation process, the compiler examines each program statement contained in the source program and checks it to ensure that it conforms to the syntax and semantics of the language. If any mistakes are discovered by the compiler during this phase, they are reported to the user and the compilation process must be started. Typical errors reported



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 13

Name Brijanka Sharma Faculty No..... Class 3rd year Batch.....

during this phase of compilation, might be due to an expression that holds unbalanced parentheses (syntax error), due to the use of a variable that is not "defined" (semantic error).

Step 3 :- When all the syntactic and semantic errors have been removed from the program and translate it into a "lower" form that is equivalent to assembly language program needed to perform the identical task.

Step 4 :- After the program has been translated the next step in the compilation process is to translate the assembly language statements into actual machine instructions. The assembler takes each assembly language statement and converts it into a binary format known as object code, which is then written into another file on the system. This file has the same name as the source file under UNIX, with the last letter an "o" (for object) instead of a "c".



M.B.M. UNIVERSITY

JODHPUR

Date _____ Experiment No. _____ P.No. 14

Name _____ Faculty No. _____ Class _____ Batch _____

Step 5: After the program has been translated into object code, it is ready to be linked. This process is once again performed automatically whenever the `cc` or `gcc` command is issued under UNIX. The purpose of the linking phase is to get the program into a final form for execution on the computer.

If the program uses other programs that were previously processed by the compiler, then during this phase the programs are linked together. Programs that are used from the system's program library are also searched and linked together with the object program during this phase.

The process of compiling and linking a program is often called building.

The final linked file, which is an executable object code format, is stored in another file on the system, ready to be run or executed. Under UNIX, this file is called `a.out` by default. Under Windows, the executable file usually has the same name as the source file, with the `C` extension replaced by an `exe` extension.



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No..... P.No. 15.....

Name..... Faculty No..... Class..... Batch.....

step 6:- To subsequently execute the program, the command, a.out has the effect of loading the program called a.out into the computer's memory and initiating its execution.

When the program is executed, each of the statements of the program is sequentially executed in turn. If the program requests any data from the user, known as input, the program temporarily suspends its execution so that the input can be entered. Or, the program might simply wait for an event, such as a mouse being clicked, to occur. Results that are displayed by the program, known as output, appear in a window, sometimes called the console. If the program does not produce the desired results, it is necessary to go back and reanalyze the program's logic. This is known as the debugging phase, during which an attempt is made to remove all the known problems or bugs from the program. To do this, it will most likely be necessary to make changes to original source program.



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No. 1 P.No. 16

Name..... Faculty No..... Class..... Batch.....

Program Name:- Write a program to print
"Hellow World".

Algorithm :-
Step 1 → Start
Step 2 → Display "Hellow World".
Step 3 → Stop

Source code:-

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

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
printf("Hellow World");
```

```
getch(); return 0
```

```
}
```

Output:-

Hellow World.



M.B.M. UNIVERSITY

JODHPUR

Date..... Experiment No. 2 P.No. 17

Name..... Faculty No..... Class..... Batch.....

Programme Name: ^{Q.1} Write a programme to add two integers.

Algorithm :-
Step 1 $\rightarrow$ Start
Step 2 $\rightarrow$ Input two no. say a & b
Step 3 $\rightarrow$ Sum = a + b
Step 4 $\rightarrow$ Display sum of two number
Step 5 $\rightarrow$ Stop

Source Code:-

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

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
int a, b;
```

```
clrscr();
```

```
printf("Enter the value of a and b \n");
```

```
scanf("%d %d", &a, &b);
```

```
printf("Sum = %d", a + b);
```

```
getch();
```

```
}
```

Output :- enter the value of a and b

16 4

SUM = 20



M.B.M. UNIVERSITY JODHPUR

Date: _____ Page: 18
Experiment No. 3 Faculty No. _____ Class _____ Batch _____

Programme Name: - Write a program to multiply two floating point numbers.

Algorithm:-
Step 1 → Start
Step 2 → Input two no. say a and b.
Step 3 → Product = $a * b$
Step 4 → Display product of two no.
Step 5 → Stop

Source code:-

```
#include <stdio.h>
#include <conio.h>
void main()
```

```
{
    float a, b;
```

~~char c;~~

```
printf("Enter the value of a and b \n");
```

```
scanf("%f %f", &a, &b);
```

```
printf("%f PRODUCT := %f \n", a * b);
```

~~getchar();~~

```
}
```

Output:- Enter the value of a and b.

4 , 5

Product := 20.000000.



M.B.M. UNIVERSITY
JODHPUR

1000

2

Abstract

References

2

11. *Journal of the American Medical Association*, 273:1233-1234 (1995).

Chapters

100

15

Programme Name: - write a program to find HCF of 2 numbers

value of a chromosome

[illegible]

Algorithm:-

step 1: $\rightarrow$ start

Step 2: $\rightarrow$ Input a character 'Day'.

Step 3: $\rightarrow$ Display ASCII value of a

Step 4 $\rightarrow$ stop

Power code: $\rightarrow$

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

include <vector>

void main()

3

cham a;

 $\phi_{\text{sea}}(t)$

Print ("Enter character: \n");

That prints
↓
single one
1/c → format specifier

8ant ($^{66}\text{N}^{29}, \text{Xa}$);

8. Count ('\\C', 'xa'),
Print ('\\ASCII Value of '\\C' = \\d', (a, a), 5)

patch();

Butterfly

output :- enter character:

ASCII value of 'e' = 101.