FUNDAMENTALS OF PROGRAMMING LANGUAGES:

UNIT I

INTRODUCTION TO PROGRAM PLANNING & C - PROGRAMMING

* HISTORY & IMPORTANCE OF C :

- C is a programming Language developed at AT & T's Bell Laboratories of USA in - 1972
- It was develop & written by man named Dennis Ritchie.
- C became more popular because it is reliable, simple & easy to use.
- In an industry where newer languages, tools & technologies emerge & vanish day in & day out, a language that has survived for more than three decades has to be really good
- Nobody can learn C++, C#, Java directly Learning these Complicated Concepts When you are not even Comfortable with basic language elements is like putting Cart before horse.
- C++, C#, Java make use of principle called Object Oriented Programming [OOP] to organize the program. But even while using this organising principle you would still need a good

hold over language elements of C & basic programming skills.

- Major parts of popular operating systems like Windows, UNIX, Linux are still written in C. This is because even today when it comes to performance (speed of execution) nothing beats C.

- If one needs to extend the operating system to work with new devices one needs to write device driver programs. These program are exclusively written in C.

- Devices like Cellular phones, palmtops, Microwave ovens, washing machine, Digital Cameras has Microprocessor, Operating system & program embedded in these devices.. This programs not only have to run fast but also have to work in limited amount of memory. No wonder that such programs are written in C.

- professional 3D computer Games where user navigates some object like say a spaceship & fires bullets at the invaders. requires great speed. To match the expectation of player, the game has to react fast to the user inputs. This is where C language scores over other languages.

Page No.
Date :

* GETTING STARTED WITH C :-

- There is close analogy between Learning English language & ~~learning~~ Learning C language.

- The classical Method of learning language is to first learn alphabets used in english language, then learn to combine these alphabets to form words, which in turn are combined to form sentences & Sentences are combined to form paragra

- In C Language, Instead of straight away learning how to write programs, we must first know what Alphabets, Numbers & Special symbols are used in C., then how using them Constants, Variables & Keywords are constructed & finally how are these combined to form an Instructio A Group of Instructions would be combinec later on to form a program.

* ALPHABETS → WORDS → SENTENCES → PARAGRA

* ALPHABETS
DIGITS
SPECIAL SYMBOLS
→ CONSTANTS
VARIABLES
KEYWORDS
→ INSTRUCTION
↓
PROGRAM

Page No.
Date :

THE C CHARACTER SET : [SPECIAL SYMBOL

- A character denotes any alphabet, digit or special symbol used to represent information.

- ALPHABETS — A, B, C ..... Y, Z
  a, b, c ..... y, z

- DIGITS — 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- SPECIAL SYMBOLS (29) — ~ ' ! @ # % ^ & * ( ) _ - + = | \ / { } [ ] : ; " " , < > , . ? $

.	period	&	ampersand
;	semi colon	^	caret
:	Colon	#	hash
'	Apostrophe		
"	double quotation	>	greater than or Closing angle brac
|	Vertical bar		
/	forward slash		
\	backward slash	<	less than or opening angle bracke
~	tilde		
$	dollar sign		
(	opening left parenthesis		
)	closing right parenthesis		

Page No.
Date :

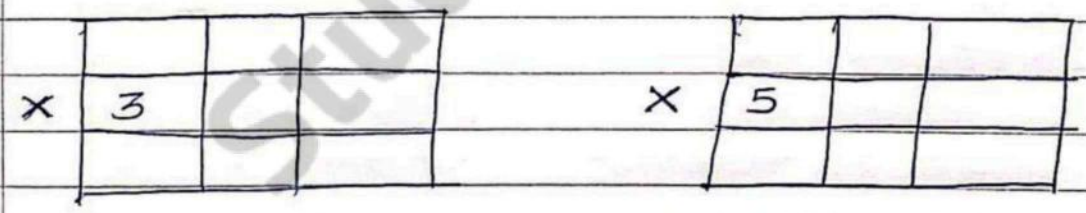
* CONSTANTS & VARIABLES

- A Constant is an entity that does not change whereas Variable is an entity that may change.

- In any program we do lots of calculations. The results of these calculations are stored in Computer's memory. The calculated values are stored in these memory cells.

- To make retrieval & usage of these values easy these memory cells are given names. Since, value stored in each location may change the names given to these locations are called Variable names.

Eg:-



- 3 is stored in memory Location & name x is given to it.
- Then we have assigned a new value 5 to same Memory location x. This would overwrite earlier value 3, since a memory location can hold only one value at a time.
- Since, the location whose name is x can hold different values at different

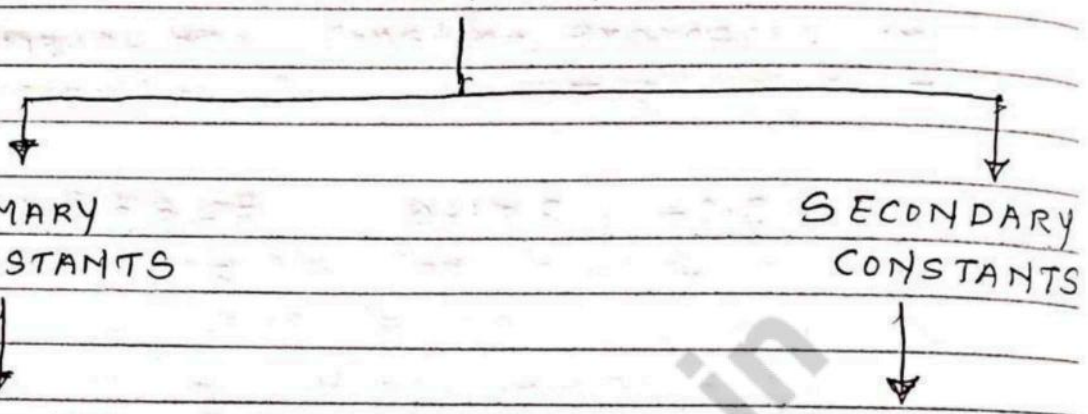
Page No.
Date :

times, x is known as Variable.

- As against this, 3 or 5 do not change hence are known as Constants.

Page No.
Date :

* TYPES OF CONSTANTS :



* PRIMARY CONSTANTS :

1. INTEGER CONSTANT

- An Integer Constant must have atleast one dig
- It must not have a decimal point.
- It can be either positive or negative.
- If no sign precedes an integer constant, it is assumed to be positive.
- No Commas or blanks are allowed within an integer Constant.
- The allowable range for integer constant is -2147483648 to +2147483647

Eg:- 426 , -8000
+782 , -7605

Page No.
Date :

2. REAL CONSTANT : / FLOATING POINT -

- They are often called Floating point Constants
- It could be written in two forms :-
Fractional form & Exponential form.

* FRACTIONAL FORM :-

- A real constant must have atleast one digit.
- It must have a decimal point.
- It could be either positive or negative.
- Default sign is positive.
- No commas or blanks are allowed within a real Constant.

Eg: +325.34 , 426.0 , -32.76 , -48.5792.

* EXPONENTIAL FORM :

- It is usually used if the value of constant is either too small or too large.
- It is represented in two parts :-
The part appearing before 'e' is called Mantissa
The part following e is called Exponent.

Eg :- 0.000342 = 3.42 e -4

- The Mantissa part & exponential part should be separated by a letter e or E.
- The Mantissa part may have positive or negative sign.
- Default sign of Mantissa part is positive.

- The exponent must have atleast one digit which must be positive or negative integer. Default sign is positive.
- Range of real constant expressed in exponential form is -3.4e38 to 3.4e38

Eg:- +3.2e-5 , 4.1e8 , -0.2E+3 , -3.2e-5.

3. CHARACTER CONSTANT :-

- A character constant is a single alphabet, a single digit or a single physical symbol enclosed within single inverted commas. Both the inverted commas should point to the left.

- Eg:- 'A' , 'I' , '5' , '='

- The Maximum length of character constant can be 1 character.

4. STRING CONSTANT :-

- It is a sequence of characters enclosed in double quote., the characters may be letters, numbers, special character & blank space.

- Eg:- "rama" , "a" , "+123" , "1-/a"

"good" // string constant

"" // null string constant.

" " // string constant of 6 white spaces

"x" // string constant having single character.

"Earth is round \n" // prints string with new line.

5. BACKLASH CHARACTER CONSTANT :-

- It is also known as Escape characters.

\n	new line
\r	Carriage return
\t	tab
\v	vertical tab
\b	backspace
\f	form feed (page feed)
\a	alert (beep)
\'	Single quote (')
\"	Double quote (")
\?	Question Mark (?)
\\	backslash (\)

* VARIABLES

- An Entity that may vary during program execution is called Variable.
- Variable names are names given to locations in memory.
- These locations can contain integer, real or character constants.
- Integer variable can hold only an integer constant, real variable can hold only a real constant & character variable can hold only a character constant.

* RULES FOR CONSTRUCTING VARIABLE NAMES

- A variable name is any combination of 1 to 31 Alphabets, digits or underscores. Do not create unnecessarily long variable names as it adds to our typing effort.
- The first character in the variable name must be an Alphabet or underscore.
- No commas or blanks are allowed within a variable name.

- No special symbol other than the underscore can be used in a variable name.

Eg:- Si_int , m_hra , pop_e_89

- C compiler is able to distinguish between the variable names by making it compulsory to us to declare the type of any variable name that you wish to use in a program.
- Examples of type declaration statement

```
int   Si , m_hra ;
float bassal ;
char  code ;
```

- If we want to calculate Simple Interest, it is always advisable to construct meaningful variable names like prin, roi, noy to represent principal, rate of interest & no. of years rather than using variables a, b, & c.

* DECLARATION OF VARIABLE :-

- A variable can be used to store value of any data type.
- Declaration of variable must be done before they are used in program.
- Variables are separated by commas & declaration statement ends with semicolon

- SYNTAX :- data_type Variable-1, Variable-2 ;

Page No.
Date:

Eg:- int x, y, z ;
float a, b ;
char m, n ;

* ASSIGNING VALUES TO VARIABLES :-
- Values can be assigned to variables using the assignment operator (=).
- SYNTAX :- Variable = constant ;
- Eg:- x = 100 ;
a = 12.25 ;
m = f ;
- We can also assign a value to a variable at the time of variable declaration.
SYNTAX :- data_type Variable = constant
Eg :- int x = 100 ;
float a = 12.25 ;
char m = f ;

Page No.
Date :

* C KEYWORDS

- They are also known as Reserved words.
- They have fixed meaning that we cannot change
- Keywords are the words whose meaning has already been explained to C compiler.
- Keywords cannot be used as Variable names because if we do so, we are trying to assign new meaning to the keyword, which is not allowed by computer.
- There are only 32 keywords available in C.

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

- We cannot use keywords as Constant name, Variable name etc.

* int main ( )

- It indicates that main func^n will return integer value.

- Eg:-
```
#include <stdio.h>
int main ( )
{
  printf ("Hello World");
  return 0;        // By default it is zero
}                  // So, if we don't write then also ok.
```

- If there is an error, it will return 1 or -1.
- return 0 tells us about exit status successfully

* Void main ( )

- void means null. It indicates that main function will not return any value to O.S.
- Eg:-
```
# include <stdio.h>
void main ( )
{
  printf ("Hello World");
}
```

* HEADER FILES :-

- First & fore most component in C-program
- It is a file with extension -h
- printf scanf - stdio.h :- defines std. input & output function
- conio.h :- defines Console input & output functi
- string.h :- defines string handling functions
- math.h :- defines common Mathematical functio
- Eg:-
```
# include <stdio.h>
# include <conio.h>
```

- Console means o/p screen functions
clrscr ( ) , getch ( ) , getche ( )

Page No.
Date :

* THE FIRST C PROGRAM :-

- Program for Calculating Simple Interest

```
/* Calculation of Simple Interest */
/* Author : girish  Date : 24/6/24 */
# include <stdio.h>
int main ( )
{
    int P, n ;
    float r, SI ;

    P = 1000 ;
    n = 3 ;
    r = 8.5 ;

    SI = P * n * r / 100 ; /* Formula for
                              simple Interest */

    printf ( " % f \n ", SI ) ;

    return 0 ;

}
```

1. Blank spaces may be Inserted between two words to improve the readability of statement. However, no blank spaces are

allowed within a variable, constant or keyword.

2. All statements are entered in small case letters.

3. Every C statement must end with ;

4. Comment about the statement must be enclosed within /* */.
Eg:- // This is one line comment.. /* Multiline coment */

5. A Comment can be written before the statement, after the statement or within the statement.

Eg:- /* formula */ SI = P*n*r/100;

SI = P*n*r/100; /* formula */

SI = P*n*r/ /* formula */ 100;

6. The text written inside a comment can be small case, capital or a combination.

7. Comments cannot be nested

Eg:- /* Cal of SI /* Author : Girish */ */

8. A comment can be split over more than one line

Eg:- /* This is
an exaggerated
comment */

9. main ( ) is a function. A function is nothing but set of statements. In C program there can be multiple functions. To begin with, we would concentrate only on those programs which have only one function. The name of this function has to be main ( ); it cannot be anything else.

All statements that belong to main ( ) are enclosed within a pair of braces { } as shown.

```
int main ( )
{
  Statement 1;
  Statement 2;
  Statement 3;

}
```

10. Some Compilers even permit us to return nothing from main ( ). In such a case, we should precede it with keyword void. Void function specifies that it returns no value. Eg:. Void main ( ).

Page No.
Date :

11. Any Variable used in program must be declared before using it.

Eg:
```
int p, n ;          /* declaration */
float r, si ;       /* declaration */

si = p* n*r /100 ;  /* usage */
```

12. In the statement, * and / are the arithematic operators. The arithematic operators available in C are +, −, *, /.

There are about 45 operators available in C. Surprisingly there is no operator for exponentiation.

13. Once the value of si is calculated, it needs to be displayed on the screen. This is achieved by using function printf().

- For us to able to use printf() function it is necessary to use # include <stdio.h at the beginning of program.
  # include is a preprocessor directive. It includes Standard i/p, o/p library functions.
- The general form of printf() function is

printf ("<format string>", <list of variables>) ;

- <format string> Can contain

%lf for double

Page No.
Date :

%f for printing real values
%d for printing integer values.
%c for printing character values.

Eg:

```
printf ("%f", SI);

printf ("%d, %d %f %f", P, n, r, SI);

printf ("simple Interest = Rs. %f", SI);

printf ("principal = %d \n Rate = %f",
                P, r);
```

Output of Last statement will be :.

```
principal = 1000
Rate = 8.500000
```

14. \n is known as new line & it takes the cursor to the next line. It is known as Escape Sequences available in 'C'.

15. printf() can not only print values of variables it can also print the result of an expression. An expression is nothing but combination of constant, variables & operators. Thus, 3, 3+2, c & a+b*c-d all are valid expressions. The results of these expression can be printed as follows :.

```
printf ("%d %d %d %d", 3, 3+2, c,
                a+b*c-d);
```

16. getch() function asks for single character until you press any key, it blocks the screen. [Holds the o/p screen until we press any key on keyboard.]

17. #include <stdio.h> It includes standard input output functions. The printf() function is defined in stdio.h

18. #include <conio.h> includes console input output library functions. The getch() function is defined in conio.h file.

19. printf() function: is used to print data on the console.

20. getch() :- It asks for single character until you press any key, it blocks the screen

Page No.
Date :

* C TOKENS :-

- The smallest individual units are known as Tokens.
- C has 6 types of tokens :-
  a) Identifiers.
  b) Keywords
  c) Constants
  d) Strings
  e) Special symbols
  f) operators

a) IDENTIFIERS :-

- It refers to names of variables, Constant, function & arrays.
- It can only have alpha-numeric Characters [a-z, A-Z, 0-9] & underscore [_]
- The first character of an identifier can only contain alphabet or underscore
- Identifiers are case sensitive.
  Eg:- name & Name are two different identifiers.
- Keywords are not allowed to use as identifiers.
- No special character, such as semicolon, period, whitespaces, slash or comma are permitted to be used as identifiers.

Page No.
Date :

Eg:- VALID	INVALID
STDNAME	Return
SUB	$stay
TOT_MARKS	1RECORD
_TEMP	STD NAME
Y2K	

Page No.
Date :

* DATA TYPES :

- A data type specifies the type of data that a variable can store such as Integer, floating, character.

1. Basic Data type :- / Primary Data type :-
Floating point, integer, double, character.

2. Derived Data type :-
Union, structure, Array.

3. Enumerated data type :-
Enums.

4. Void Data type :-
Empty value

5. Bool type :-
True or False.

1] PRIMARY DATA TYPE :-

- INTEGER :- is used for storing various whole numbers such as 5, 8, 67, -3, 2390
Eg :- int 5 ; . int b ;

2] • CHARACTER :-
It stores a single character belonging to defined character set of C Language.
Eg :- char a ;

- FLOATING POINT :-
These refers to all real number values

[precision of 6 digits after decimal points.]

or decimal points such as 40.1, 820.673, 5.9 etc. Eg:- float c;

- DOUBLE
- It includes all large type of numeric values that do not come under either floating type data or integer
- It is capable of holding double the size of info. & data as compared to that of float.
- It is capable of holding about 15 to 16 digits after & before any given decimal point.
- Eg:- 1.9876, 5.43219, -876.543, 21.987654, 0.15197 e-7.
  double c;

- VOID :-
- This term refers to no values at all.
- It is used to define functions in a program.

* DATA TYPE MODIFIERS :-

- Modifiers in C-Language help in making primary data type much more specific.
- There are basically 4 types of Modifiers :-
  - UNSIGNED :- only positive.
  - SIGNED :- positive & negative
  - SHORT } Cause an effect on the value
  - LONG } range of any given data type.

Page No.
Date :

* SIZE & RANGE OF DATA TYPES :

	TYPE	SIZE (bytes)	RANGE	CONTROL STRING.
1.	Signed char.	1	-128 to 127	% c
2.	Unsigned char	1	0 to 255	% c
3.	Signed int	2	-32768 to 32767	% d or % i
4.	Unsigned int	2	0 to 65535	% u
5.	Signed short int	1	-128 to 127	% d or % i
6.	Unsigned short int	1	0 to 255	% d or % i
7.	Signed long int	4	-2147483648 to 2147483647	% ld
8.	Unsigned long int	4	0 to 4294967295	% lu
9.	float	4	3.4E-38 to 3.4E+38	% f or % g
10.	Double	8	1.7E-308 to 1.7E+308	% lf
11.	Long double	10	3.4E-4932 to 1.1E+4932	% Lf

Page No.
Date :

* STORAGE CLASS :

- In C Language, each variable has a storage class which is used to define scope & life time of a variable.

- Any variable declared in program can be stored either in memory or registers.

- Registers are small amount of storage in CPU. The data stored in registers has fast access compared to data stored in memory.

- It gives information about location of variable in which it is stored, initial value of variable if storage class is not specified, Scope of the variable, life of the variable.

1. AUTOMATIC STORAGE CLASS :

- It is the default storage class in C.
- To define a variable as automatic storage class, keyword "auto" is used.

- Scope of the variable is within the block where it is defined & life of the variable is until the control remains within the block.

- SYNTAX

```
auto data_type Variable_name;
auto int a, b;
```

Page No.
Date :

Eg:-

```
Void main ( )
{
    int detail;
      or
    auto int detail;    // Both are same
}
```

2. REGISTER STORAGE CLASS:

- To define the variable as register storage class, the keyword 'register' is used.
- When variable is declared as register, it is stored in the CPU registers.
- Register variable has faster access than normal variable.
- Frequently used variables are kept in registers.
- Only few variables can be placed inside the register.
- SYNTAX

```
register data_type Variable_name;
```

Eg:- `register int i;`

Page No.
Date :

3. STATIC STORAGE CLASS :-

- When the variable is declared as static it is stored in the memory.
- The default value of the variable will be zero.
- To define a variable as static storage class, the keyword 'static' is used.
- A static variable can be initialised only once, it cannot be reinitialised.

- SYNTAX :-

```
Static data_type variable_name;
```

Eg:- `Static int i;`

Page No.
Date :

4. EXTERNAL STORAGE CLASS

- When the variable is declared as extern, it is stored in the memory.
- The default value is initialised to zero
- The scope of variable is global & life of the variable is until program execution comes to an end.

- To define the variable as external storage class, the keyword "extern" is used.
- An extern variable is also called as global variable.
- It remains available throughout the entire program.
- Their values can be changed by any function in the program.

- SYNTAX :- extern data_type Variable_name ;
  Eg :- extern int i ;

Page No.
Date :

STORAGE CLASS	STORAGE PLACE	DEFAULT VALUE	SCOPE	LIFE TIME
auto	RAM	Garbage value	Local	within function
extern	RAM	Zero	Global	Till end of Main program. May be declared anywhere in program
static	RAM	Zero	Local	Till the end of Main program. Retains value betⁿ multiple funcⁿ call.
register	Register	Garbage Value	Local	Within funcⁿ

Page No.
Date :

* DECLARING VARIABLE AS CONSTANT :-

1. USING const KEYWORD :-

- The const keyword specifies that a variable or object value is constant & can't be modified at the compilation time.

- SYNTAX :-

```
const data_type variable_name = initial_value;
```

Eg:-

```
# include <stdio.h>

int main ()
{
    const int num = 1;

    num = 5;        // Modifying the value.

    return 0;
}
```

OUTPUT :- It will throw an error as variable value is constant & cannot be modified.

2 USING enum KEYWORD :-

- Enumeration is user defined data type in C.
- It is mainly used to assign names to integral constants, that make a program easy to read & maintain.

- It gives an opportunity to invent our own data type & define what values the variable of this data type can take.
- It helps to reduce programming errors.
- It helps in making program listing more readable, which can be an advantage when program gets complicated or when more than one programmer would be working on it.

Eg:- We can invent a data type called mar_status which can have 4 possible values - single, married, divorced or widowed.

```
# include <stdio.h>

enum VARS { var = 42 };

// where mytype = int, char, long but it
can't be float, double or user defined data type

int main ()
{
   printf ("The value of var : %d", var);

   return 0;

}
```

OUTPUT : The value of var : 42.