Practical-1

Programming Language

- A programming language is a set of commands, instructions,and other syntax use to create a software program.
- Languages that programmers use to write code are called "high-level languages."

Programming Language

- Programming Languages are of two types:
  - Low Level Programming Language : Low-level languages include assembly and machine languages.
  - High Level Programming Language : High-level languages are designed to be easy to read and understand.
  - Examples of high-level languages include C, C++, Java, Perl, and PHP

C Language/PPS

- C Is A High-Level Programming Language Used To Create High Level Program.
- It Is A General Purpose And Structured Programming.
- Developed By Dennis Ritchie At AT & T Bell Laboratories In 1972 In USA.
- It Is Also Called As Procedure Oriented Programming Language.
- It Is So Popular Language Because It Is Reliable, Simple And Easy To Use.

C Language

- It Is Well Suited For Business And Scientific Applications.
- It Has Some Various Features Like Control Structures, Looping Statements, Arrays, Macros Etc.
- C Language Is Called "Compiled Languages" Since The Source Code Must First Be Compiled In Order To Run.

Evolution/History of C Language

- An International Committee Developed ALGOL 60 Language Which Is Used To Program All Type Of Applications Such As Commercial Applications, Scientific Applications, And System Applications And So On.
- ALGOL 60 Never Became Popular Because It Was Too Abstract And Too General.
- To Reduce This Abstractness And Generality, A New Language Called Combined Programming Language (CPL) Was Developed At Cambridge University.
- CPL Turned Out To Be So Big, Having So Many Features, This It Was Hard To Learn And Difficult To Implement.

Evolution/History of C Languge

- Basic Combined Programming Language (BCPL), developed by Martin Richards at Cambridge University to solve the problems of CPL.
- It is less powerful and too specific.
- Around same time a language called “B” was written by Ken Thompson.
- But like BCPL, B is also too specific.
- Finally Dennis Ritchie inherited the features of B and BCPL, added some of his own stuff and developed “C”.

Features of C Language

- Robust language, which can be used to write any complex program.
- Well-suited for writing both system software and business applications.
- Dynamic memory allocation C is highly portable.
- This means that ‘C’ programs written from one computer can be run on another computer with no modification.
- A ‘C’ program is basically a collection of functions that are supported by the ‘C’ library.

Features of C Language

- Program written in C language are efficient & fast.
- ‘C’ is a free form language.
- Case sensitive
- C has 32 keywords.
- ‘C’ is a structure or procedural programming language that are use top-down approach.

Basic structure of C program

- Structure of C program is defined by set of rules called protocol, to be followed by programmer while writing C program.

Documentation Section

- The documentation section is the part of the program where the programmer gives the details associated with the program.
- Example
- **Example**

```
//File Name: Hello.c
/*
  Author: ABC
  date: 23/03/2021
  description: a program to display hello world  */
```

Link Section

- Header files that are required to execute a C program are included in this section
- This leads to the compiler being told to link the header files to the system libraries.
- Example
  - #include <stdio.h>

Definition Section

- Define variable and set value for variable.
- we define different constants.
- The keyword define is used in this part.
- Example
  - #define PI=3.14

Global Declaration Section

- Global variables are defined in this section.
- When a variable is to be used throughout the program, can be defined in this section.
- The user-defined functions are also declared in this part of the code.
- Example
  - Int a=10
  - Int age(int b);

Main Function Section

- Every C-programs needs to have the main function.
- Each main function contains 2 parts.
- A declaration part and an Execution part.
- The declaration part is the part where all the variables are declared.
- The execution part begins with the curly brackets and ends with the curly close bracket.

Example

```
Void main()
{
Int a;
Float b;
printf(“%d %f”,a,b);
getch()
}
```

Sub Program Section

- All the user-defined functions are defined in this section of the program.
- User can define their own functions in this section which perform particular task as per the user requirement.
- Example

```
Int addition(int a ,int b)
{
return a+b;
}
```

Commonly used escape sequences are

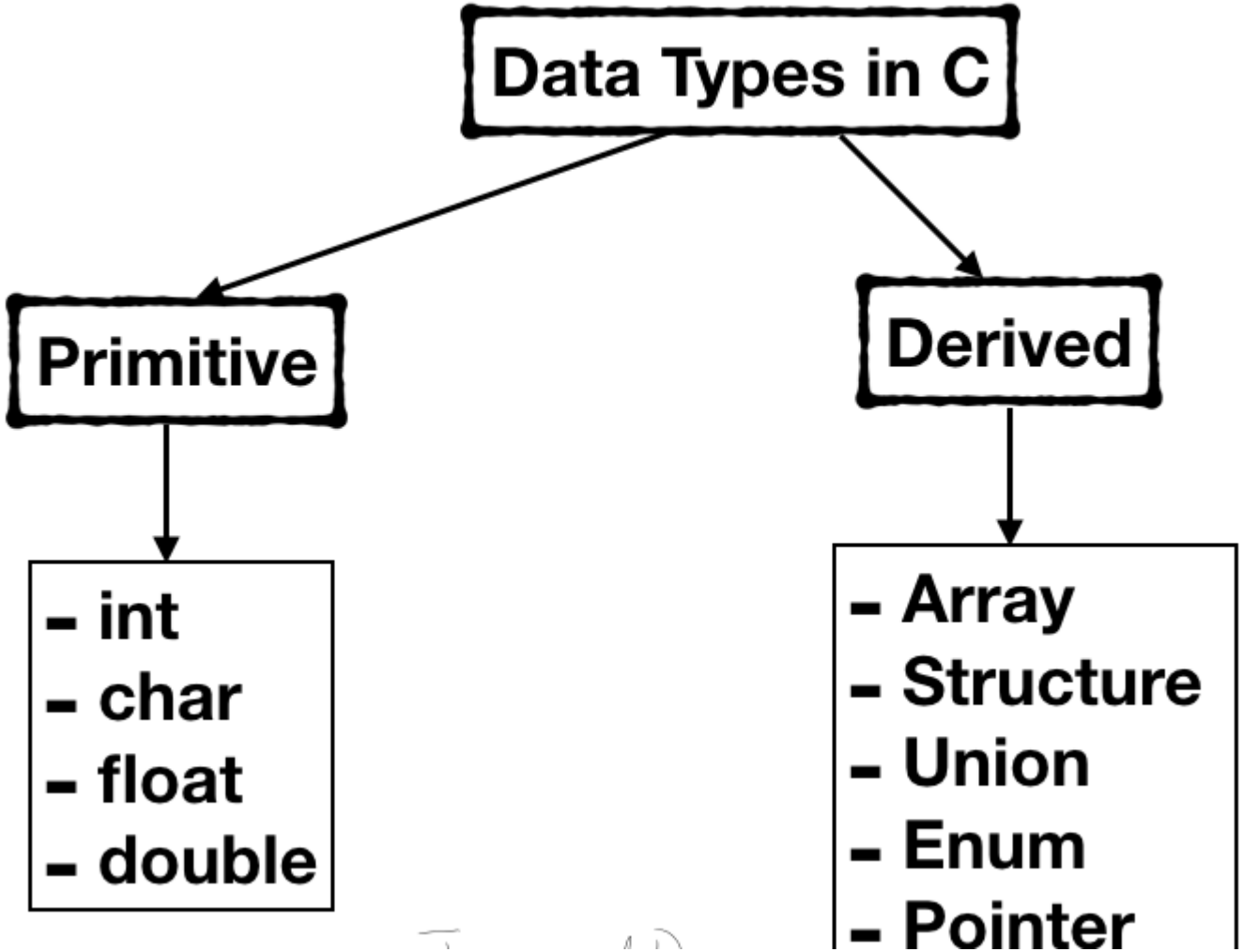
\a	Alarm or Beep
\b	Backspace
\f	Form Feed
\n	New Line
\r	Carriage Return
\t	Tab (Horizontal)
\v	Vertical Tab
\\	Backslash
\'	Single Quote
\"	Double Quote
\?	Question Mark
\ooo	octal number
\xhh	hexadecimal number
\0	Null

Practical-1.2

Write a single program to print following messages using single as well as multiple printf() statements:

Hello World	********** * Hello World* **********	"15/8/1947"	Name: <your name> Phone No: <mobile> Address: <Full address> Birth Date: <dd/mm/yyyy>
"Hello World"	Hello World	15\\8\\1947	

Practical-1.3
Data Types



Practical-1.3
Format Specifiers

Data Type		Format
Integer	Integer	%d
	Short	%d
	Short unsigned	%u
	Long	%ld
	Long assigned	%lu
	Hexadecimal	%x
	Long hexadecimal	%lx
	Octal	%O (letter 0)
	long octal	%lo
Real	float,double	%f, %lf, %g
Character		%c
String		%s

Practical-1.3

Write a program to take two input values x and y from keyboard and print the following as output (Hint: use \t for alignment):

```
                    x + y = answer

x - y = answer      x % y = answer      x * y = answer

                    x / y = answer
```

Practical-1.4

Write a program which takes a kilometer value as input and display in following converted units:

```
Enter distance in kilometers: 2
The distance in Feet: 6561.680176
The distance in Inches: 78740.156250
The distance in Meters: 2000.000000
The distance in Centimeters: 200000.00000
```

Practical-1.5

Write a program to swap (interchange) values of two variables with and without using third variable.

Practical-1.5

STEP 1: START.

STEP **2**: ENTER x=8 y=9

STEP 3: PRINT x, y.

STEP 4: x = x + y x=17

STEP 5: y= x - y. 17-9=8

STEP 6: x =x - y. 17-8=9

STEP 7: PRINT x, y.

STEP 8: END.

STEP 1: START.

STEP **2**: ENTER x, y.

STEP 3: PRINT x, y.

STEP 4: Temp= x

STEP 5: x= y

STEP 6: y=Temp

STEP 7: PRINT x, y.

STEP 8: END.