Paper CH-IV Analytical Chemistry –I

UNIT-IV

Computational Chemistry:

Computers in chemistry: Hardware and software's: data representations, flow chart and writing simple programs in FORTRAN and c-languages e.g. solving quadratic equation, least square fitting, and titration curves etc.

Use of excel for data fitting and calculations.

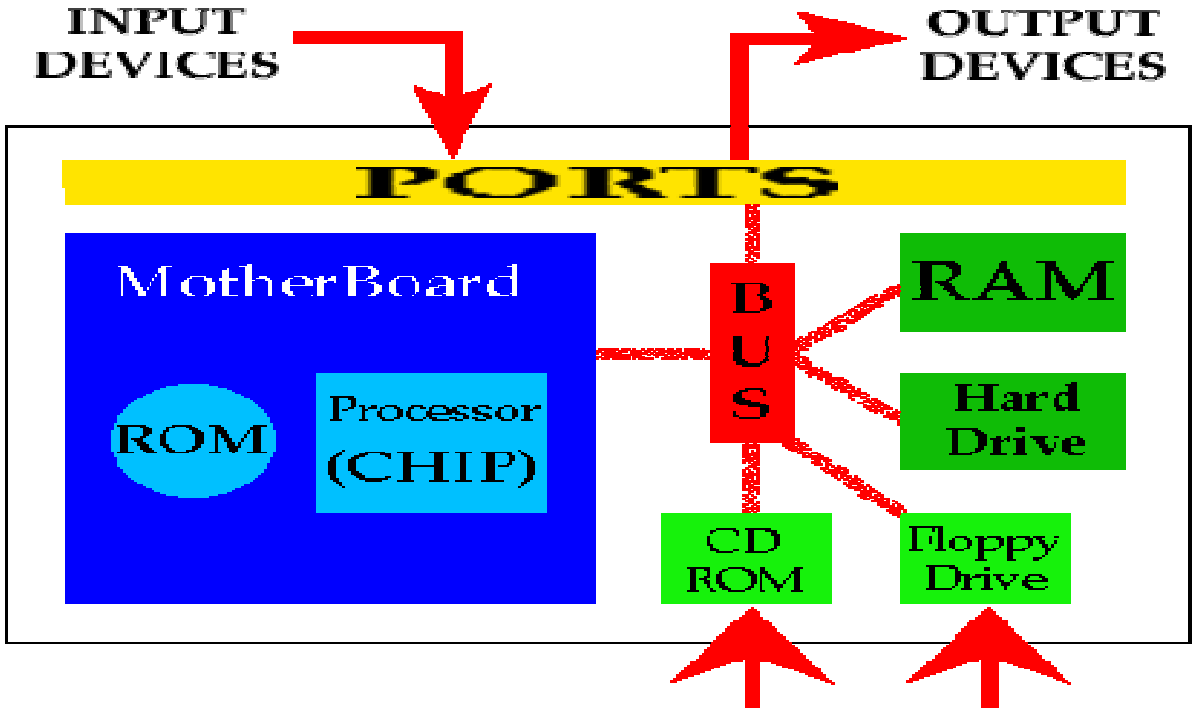
- Starts with the very basic concepts of operating systems, DOS , windows , Linux
- Presents elementary programming concepts and statements using the BASIC programming languages
- Handling graphics, and numerical methods
- Includes many solved examples from physical, organic, and inorganic chemistry
- Assumes no prior knowledge of computers or programming

Basic Concept of OS-

Each computer system includes a basic set of programs called the *operating system*. The most important program in the set is called the *kernel*.

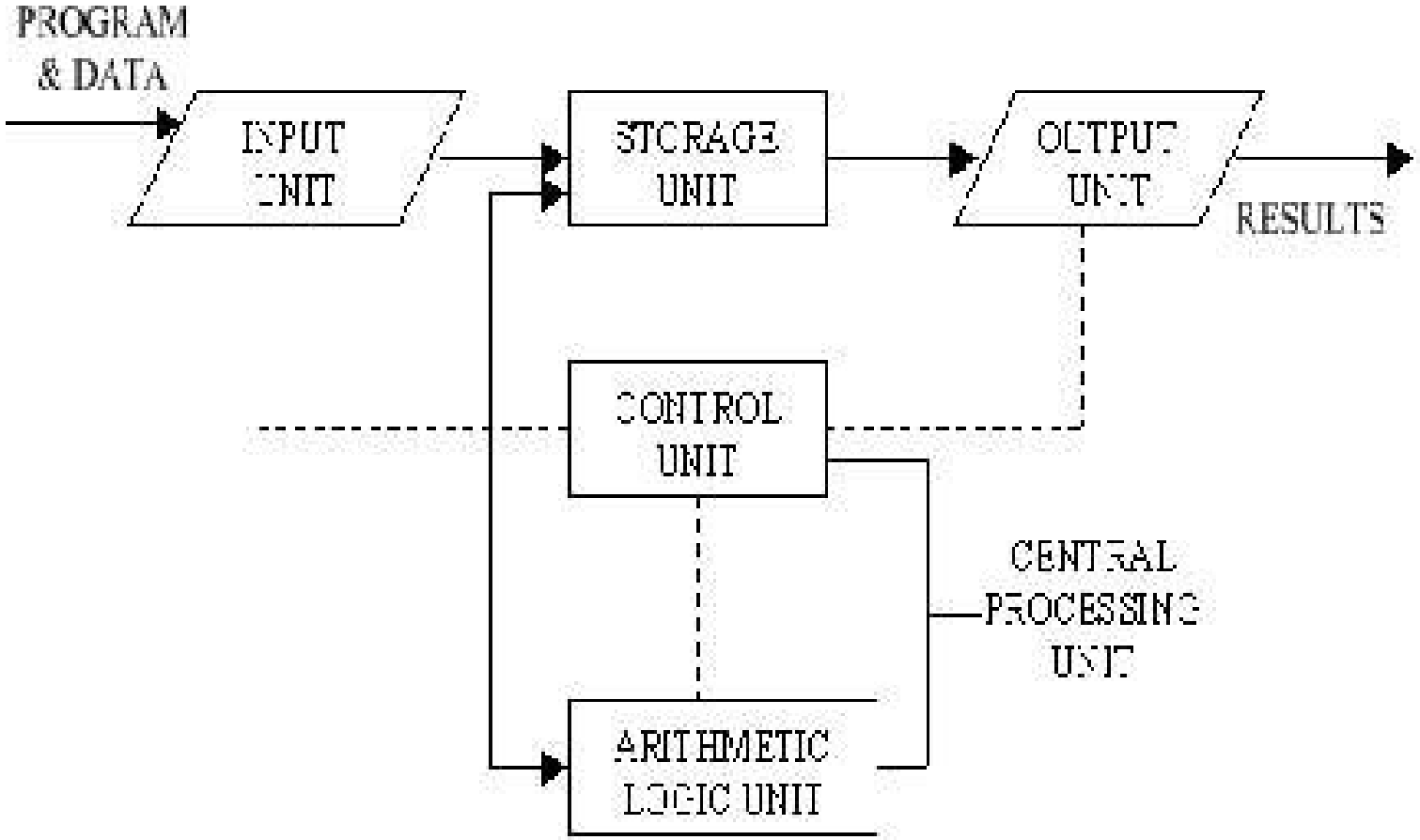
The operating system must fulfill two main objectives:

- Interact with the hardware components, servicing all low-level programmable elements included in the hardware platform.(Hardware-Parts of a computer)
- Provide an execution environment to the applications that run on the computer system (the so-called user programs). Set of programs is called software



The Central Processing Unit: (CPU), Buses, Ports and controllers, ROM; Main Memory (RAM);
Input Devices; Output Devices;
Secondary Storage; floppy disks, hard disk, CD-ROM

Block Diagram-



Data Representation

Data Representation refers to the methods used internally to represent information stored in a computer. Computers store lots of different types of information:

numbers,text,graphics (animation),sound

Memory consists of bits (0 or 1)

bytes (=8 bits)

Numbers are represented using various number systems and for text codes are available-ASCII

Number Systems -

- **Binary Number System-**
- Binary is another way of saying Base Two. So, in a binary number system, there are only two symbols used to represent numbers: 0 and 1. When we count up from zero in binary, we run out of symbols much more frequently.
- 0, 1, …
- From here, there are no more symbols. We do not go to 2 because in binary, a 2 doesn't exist.

- **Binary**
- 0 1 10 11 100 101 110 111 1000 1001 1010
- **Decimal**
- 0 1 2 3 4 5 6 7 8 9 10
- Just like in decimal, we know that the more digits there are, the larger the number. However, in binary, we use powers of two. In the binary number 1001101, we can create a chart to find out what this really means.
- $\mathbf{2^6}$ $\mathbf{2^5}$ $\mathbf{2^4}$ $\mathbf{2^3}$ $\mathbf{2^2}$ $\mathbf{2^1}$ $\mathbf{2^0}$
- 1 0 0 1 1 0 1
- 64 +0+ 0+ 8+ 4+ 0 +1= 87

Octal Number System-

- Octal is another number system with less symbols to use than our conventional number system. Octal is fancy for Base Eight meaning eight symbols are used to represent all the quantities. They are 0, 1, 2, 3, 4, 5, 6, and 7.
- **Octal**
- 0 1 2 3 4 5 ------ 10 11
- **Decimal**
- 0 1 2 3 4 5 -------- 8 9
- $\mathbf{8^2}$ $\mathbf{8^1}$ $\mathbf{8^0}$
- 2 3
- 16+3=19

Hexadecimal Number System-

- The hexadecimal system is Base Sixteen. As its base implies, this number system uses sixteen symbols to represent numbers. Unlike binary and octal, hexadecimal has six additional symbols that it uses beyond the conventional ones found in decimal. But what comes after 9? 10 is not a single digit but two… Fortunately, the convention is that once additional symbols are needed beyond the normal ten, letters are to be used. So, in hexadecimal, the total list of symbols to use is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.
- **Hexadecimal**
- 9 A B C D E F
- **Decimal**
- 9 10 11 12 13 14 15
- **16^3 16^2 16^1 16^0**
- A 7 =167
- 10*16+7*1 =167

Problem solving on a computer-

- It involves four steps

Step-1 Writing algorithms :

Algorithm can be defined as: “A sequence of activities to be processed for getting desired output from a given input.”

- Example of Algorithm

Problem 1: Find the area of a Circle of radius r.

Inputs to the algorithm: Radius ‘ r’ of the Circle.

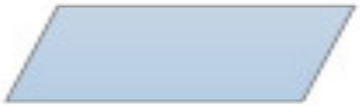
Expected output: Area of the Circle

- Algorithm
- Step 1: Start
- Step2: Read input the Radius r of the Circle
- Step3: calculation of area (Area= 3.14 r*r)
- Step4: Print Area
- Step5: Stop

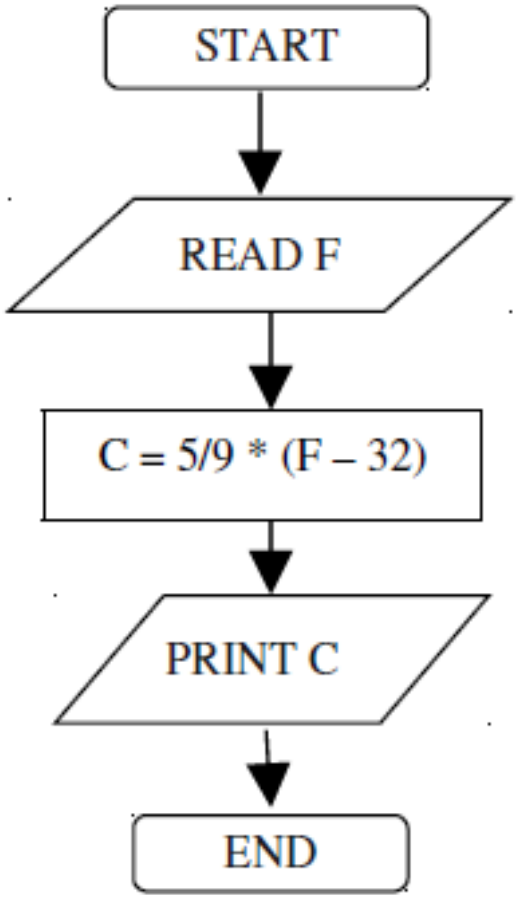
Step-2 Draw flowcharts

Diagrammatical representation of an algorithm .

Basic Symbols

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Example . Convert temperature Fahrenheit to Celsius.



Step-3 Developing code (program)using high level languages

Different high level programming languages are available for coding such as BASIC, FORTRAM, PASCAL .'C'. Use any one and convert your flowchart to that language using its coding structure and syntax rules and save this code as source code file in a computer.

Step-4 Translating and running the code(program)

- Once your source code is ready than translate it using language processor (compiler or interpreter) as object code and run it to get desired result.

Problem Statement

- Given a quadratic equation as follows:

$$ax^2 + bx + c = 0$$

- if **b*b-4*a*c** is non-negative, the roots of the equation can be solved with the following formulae:

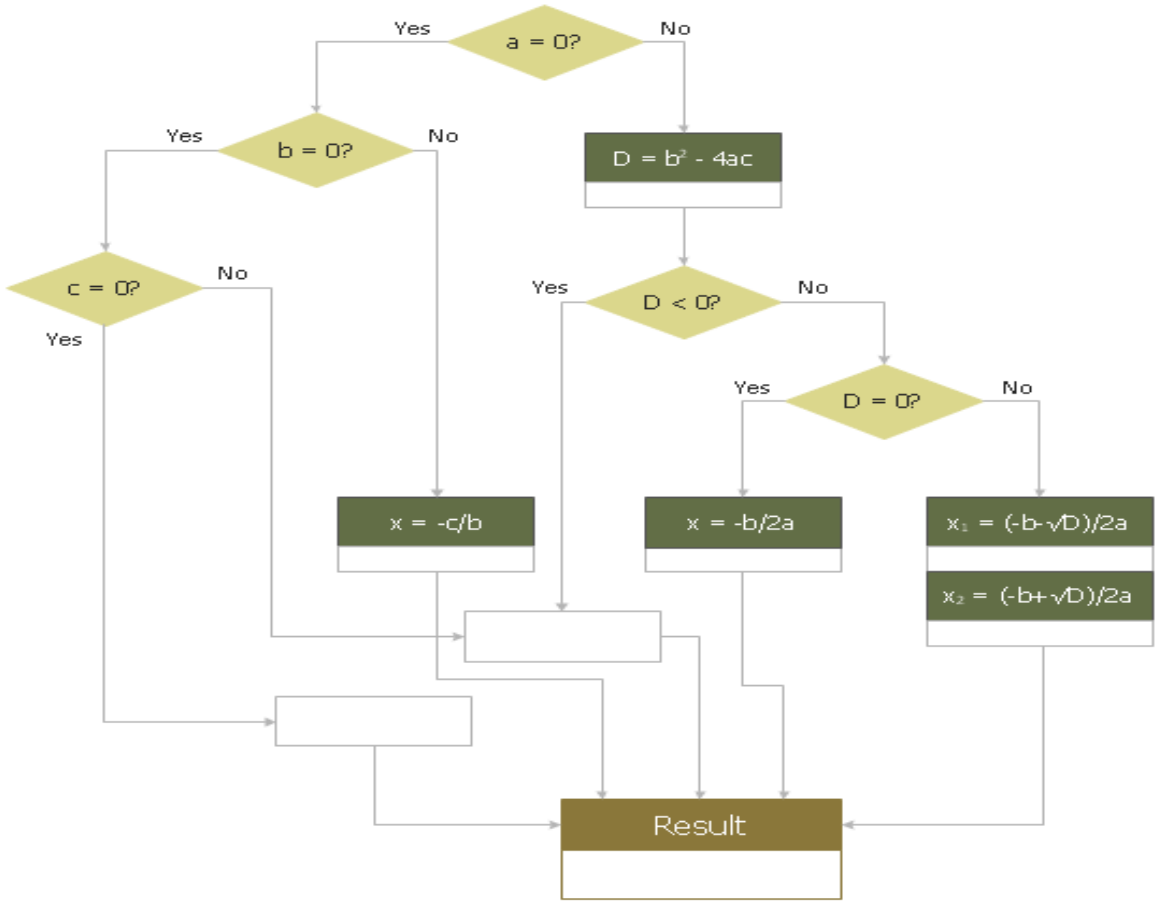
$$\text{root } 1 = \frac{1}{2a}\left(-b + \sqrt{b^2 - 4ac}\right)$$
$$\text{root } 2 = \frac{1}{2a}\left(-b - \sqrt{b^2 - 4ac}\right)$$

Algorithm for Solving Quadratic Equations

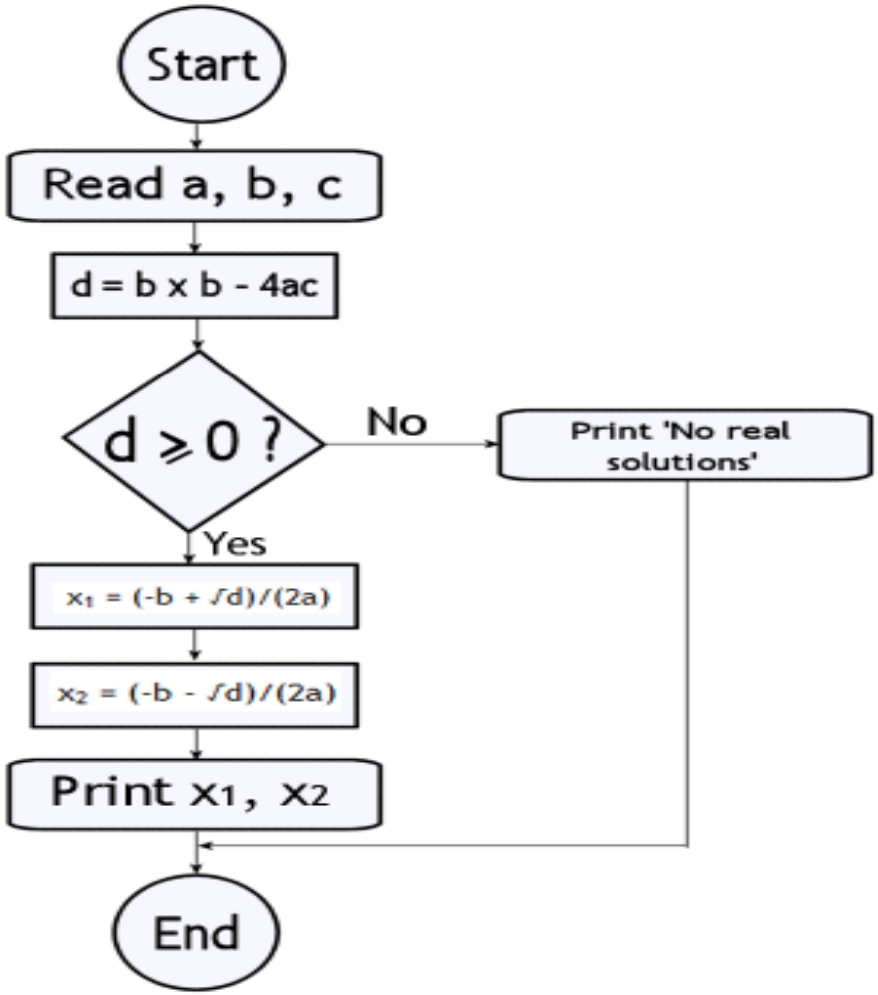
$$ax^2 + bx + c = 0$$

Enter the values for a, b, c and see the result

a = b = c =



Flow chart of solving quadratic equation



Coding Using FORTRAN-

FORTRAN is a high level language . FORTRAN stands for formula Translation . It was defined and written to translate it by John Backus of IBM and his group 1956-57.

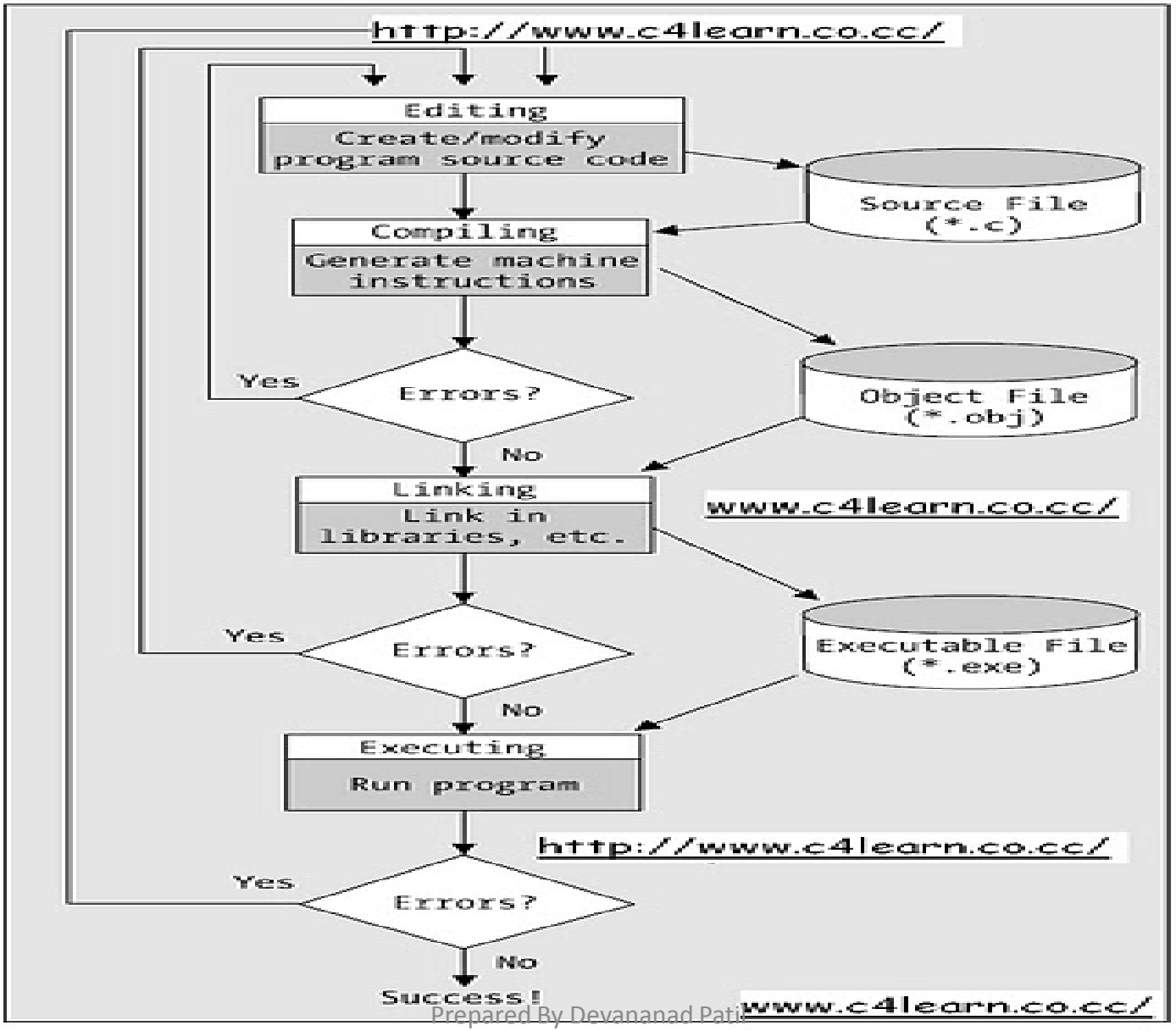
- **features**
- **Simple to learn** - when FORTRAN was design one of the objectives was to write a language that was easy to learn and understand.
- **Machine Independent** - allows for easy transportation of a program from one machine to another.
- **More natural ways to express mathematical functions** - FORTRAN permits even severely complex mathematical functions to be expressed similarly to regular algebraic notation.
- **Problem orientated language** –Procedures and Functions

FORTRAN Coding Form – Used to format source code, a carryover from IBM punch cards

```
Program -     Problem title                                          Date : 7/10/2016
        Coded by – Name of programmer                          Page –  -- of ---
                              Checked by --
1---5 6 7 ----------------------------------------------------------------------------72 73-----80
c--------------DISPLAY KRP-------------------------------------------------------------------
---------PRINT   *,'KRP'-------------------------------------------------------------------
      -----STOP   ----------------------------------------------------------------------
            END
```

1-5 comments must begin with a ***** or **C** or **!** in column 1
statement labels must occur in columns 1-5
continuation lines must have a non-blank character in column 6
statements must start in column 7
the line-length may be limited to 72 characters (derived from the 80-byte width of a punch-card, with last 8 characters reserved for (optional) sequence numbers)
Column 73 to 80 are for identification

Compilation and Running a High level program-



Checking Errors

During Compilation Compiler will check for error, If compiler finds any error then it will report it. These are **syntax errors.** User have **to re-edit the program.**

After **re-editing program , Compiler again check for any error**. If program is error-free then program is linked with appropriate libraries.

- If run time error occurs then “Run-time” errors are reported to user . if we get wrong results due to logic fault then these errors are treated as **logical errors.**
- Again programmer have to review code and check for the solution.

-

Sample program-

To find area of a circle

```
1-------7------------------
C AREA OF A  CIRCLE
      REAL R,A
      READ * ,R
      A = 3.14 * R * R
      PRINT *, R, A
     STOP------
      END----------
```

```
C    PROGRAM  Quadratic Equation
      REAL  a, b, c
      REAL  d
      REAL  root1, root2
C  read in the coefficients a, b and c
READ(*,*)  a, b, c
WRITE(*,*) 'a = ', a
WRITE(*,*) 'b = ', b
WRITE(*,*) 'c = ', c
WRITE(*,*)
 C  compute the square root of discriminate d
d = b*b - 4.0*a*c
IF (d >= 0.0) THEN
            d     = SQRT(d)
root1 = (-b + d)/(2.0*a)
root2 = (-b - d)/(2.0*a)
WRITE(*,*)  'Roots ar', root1, root2
ELSE
WRITE(*,*)  'There is no real roots!'
END IF
END
```