Unit II
Structuring the Data, Computations and Program

Elementary Data Types : Primitive data Types, Character String types, User Defined Ordinal Types, Array types, Associative Arrays, Record Types, Union Types, Pointer and reference Type.
Expression and Assignment Statements: Arithmetic expression, Overloaded Operators, Type conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed mode Assignment.
Statement level Control Statements: Selection Statements, Iterative Statements, Unconditional Branching.
Subprograms: Fundamentals of Sub Programs, Design Issues for Subprograms, Local referencing Environments, Parameter passing methods.
Abstract Data Types and Encapsulation Construct: Design issues for Abstraction, Parameterized Abstract Data types, Encapsulation Constructs, Naming Encapsulations.

Statement level Control Statements

Statement level Control Statements:

- Two linguistic mechanisms which are necessary to make the computations in programs flexible and powerful: some means of **selecting among alternative control flow paths (of statement execution)** and **some means of causing the repeated execution** of statements or sequences of statements.
- Statements that provide these kinds of capabilities are called **control statements**.
- It was proven that all algorithms that can be expressed by flowcharts can be coded in a programming language with only two control statements: one for choosing between two control flow paths and one for logically controlled iterations (**Böhm and Jacopini, 1966**).
- A **control structure** is a control statement and the collection of statements whose execution it controls.

```
e.g.
if(a>b)
{
---
---
}
else
{
---
---
}
```

Unit II
Structuring the Data, Computations and Program

1. Selection Statements

A selection statement provides the means of choosing between two or more execution paths in a program.

e.g.

simple if

if else

Nested if

else if ladder

switch

2. Iterative Statements

- An iterative statement is one that causes a statement or collection of statements to be executed zero, one, or more times.
- An iterative statement is often called a loop.

e.g.

for

while

do while

3. Unconditional Branching

- An unconditional branch statement transfers execution control to a specified location in the program.
- Without restrictions on use, imposed by either language design or programming standards, goto statements can make programs very difficult to read, and as a result, highly unreliable and costly to maintain.
- A few languages have been designed without a goto for example, Java, Python, and Ruby.
- The relatively new language, C#, includes a goto, even though one of the languages on which it is based, Java, does not.

e.g. goto statement