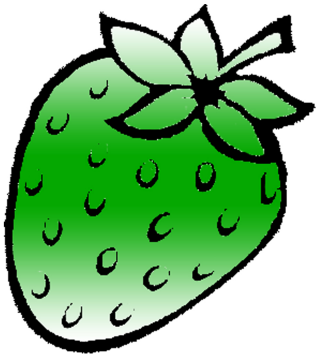
STRAWBERRY



/strawberrydevelopers

/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com

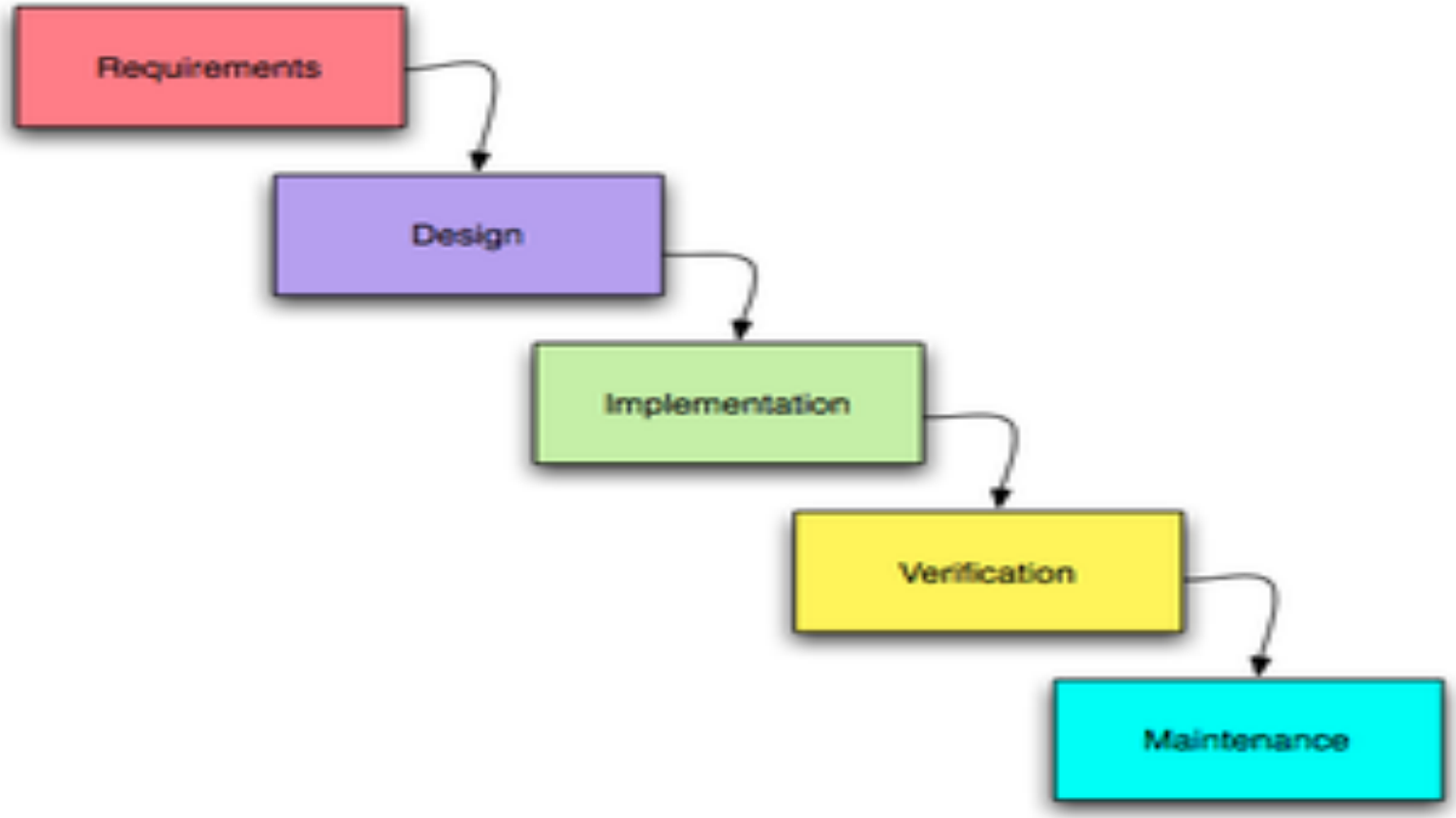
Computer Programming-I

Objective of CP-I

The course will enable the students to understand the basic concepts of structured programming.

What is programming?

- Writing a set of instructions that computer use to perform specific operations.



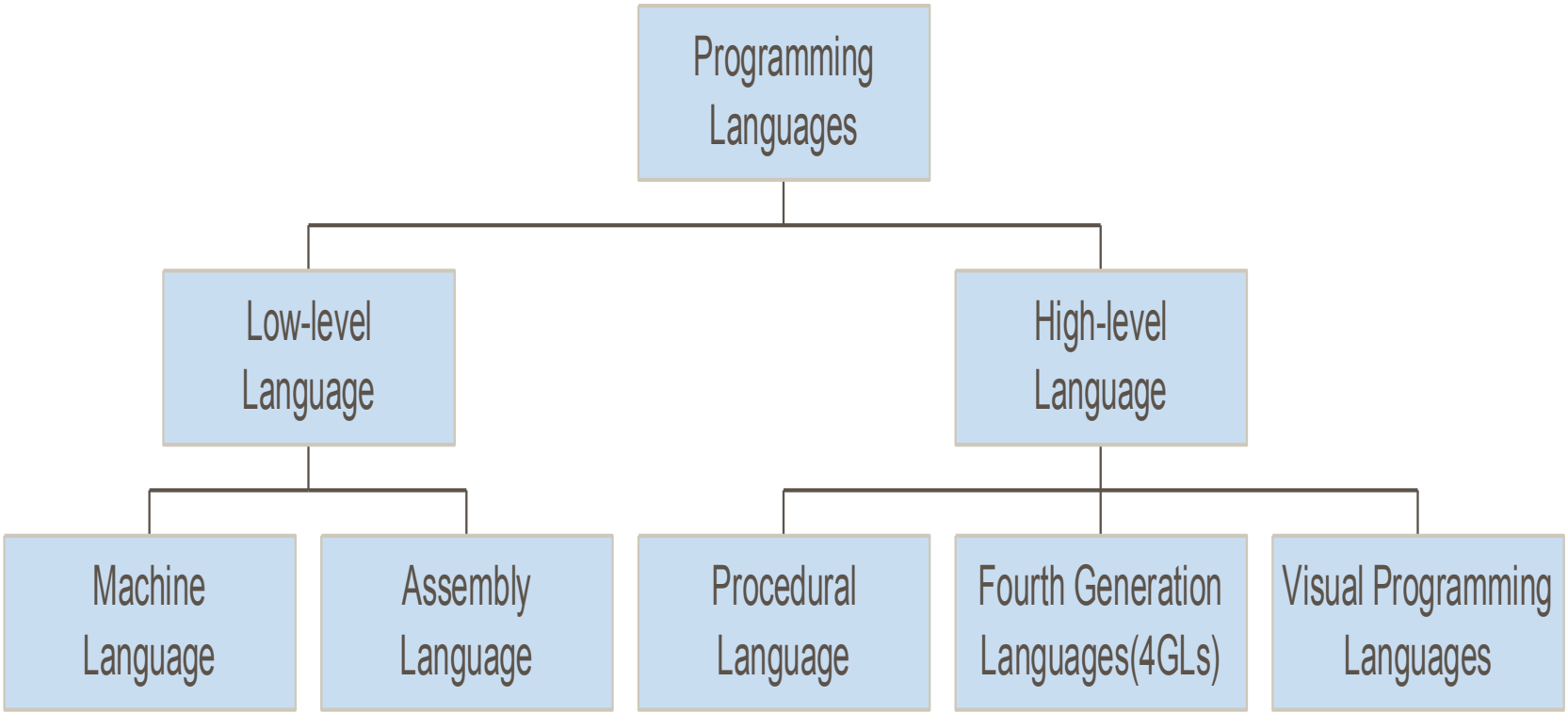
What is programming language?

- Set of instructions are written in a programming language.
- Need for programming language – to communicate instructions to machine (computer).

What is Programming ?

- Art of solving computational problems by computer.
- Computer is an electronic device and does not understand natural language.
- Set of symbols, characters, grammar rules that permit people to construct instructions in the format that can be interpreted by the computer system.

Language Types



Language Generations

On the basis of development, programming languages can be divided into 5 generations :

First Generation Language	Machine Language (1940 – 1950)
Second Generation Language	Assembly Language (1950 – 1958)
Third Generation Language	Procedural Languages (1958 – 1985)
Fourth Generation Language	4GLs (1985 onwards)
Fifth Generation Language	Visual/Graphic Languages (1990 onwards)

Machine Level Language (1940-50)

- Language containing binary code (1,0) which the computer can understand.
- Instruction contains two parts:
  - Operation Part – Specifies what is to be performed.
  - Address Part – Specifies the location of data to be manipulated.

Advantages-Machine Language

- Directly executable.
- Most efficient use of computer system resources like storage, register, etc.
- Can be used to manipulate individual bits.

Disadvantages-Machine Language

- Not portable as device dependent.
- More error prone and difficult to debug.
- Storage locations have to be addressed directly, and not symbolically.
- Increases programmer training cost as requires a high level of programming skills.
- Requires knowledge of the computer's intricate details.

Assembly Language (1950-58)

- Substitute alphabetic symbols for the binary codes of machine language.
- Symbols – memory locations.
- Mnemonics – operation code.
- One-to-one correspondence between assembly & machine language.
- Requires an assembler to convert assembly language into machine language.

- MVI B, 06 //Load Register B with the Hex value 06
- MOV A, B //Move the value in B to the Accumulator or register A
- MVI C, 07 //Load the Register C with the second number 07
- ADD C //Add the content of the Accumulator to the Register C
- STA 8200 //Store the output at a memory location e.g. 8200
- HLT //Stop the program execution

Advantages-Assembly Language

- Because symbols are meaningful, it is easier to read and understand.
- Relieves users of the problems in allocating computer storage.
- Encourages modular programming.
- Used only when efficiency is the must or when there is a need to manipulate processor registers, signals, etc.

Disadvantages-Assembly Language

- Machine-dependent and hence not portable.
- Knowledge of details of logical structure of the computer.
- Writing is difficult & time-consuming.
- Requires rigorous training.
- Not directly executable, require Assembler.
- One-to-one correspondence with machine language.

Assembler

- A program to translate an assembly program (source code) into its machine equivalent(object code).

- Procedure :
  - After the object program is created, it is transferred into the computer's primary memory using the system's loader.

  - Another program called link editor passes computer control to the first instruction in the object program, and then the execution starts and proceeds till the end of the program.

High Level languages (1958….)

- They are machine-independent as they relate to the procedures being coded.
- A HLL program can be executed on any computer system that has a translator for that HLL.
- Translated into machine code by compilers & interpreters.
- Written in English-like language.

Advantages-HLL

- Machine-independent hence portable.
- Easier to learn & requires less time to code.
- Provides better documentation.
- Libraries of subroutines can be incorporated and used in many other programs.
- Easier to debug as translators display all errors with proper error messages.

Some High Level Languages

FORTRAN	Engineering and Scientific work
COBOL	Business data processing
BASIC	Learnt quickly by beginners, popular among users of small computers
PASCAL	Used in teaching computer programming, useful in system programs because of rich data structure representation.
SNOBOL	Used in Symbol manipulation
LISP	Solving logical complex problems (chess, prove theorems)
ADA	Complex military applications

Translators

INTERPRETER	COMPILER
Translates the program line by line.	Translates the entire program.
Each time the program is executed, every line is checked for syntax and then converted to equivalent machine code.	Converts the entire program to machine-code, when all the syntax errors are removed, and executes the object code directly.
Source program & the interpreter are required for execution.	Neither source nor the compiler are required for execution.
Good fro fast debugging and at testing stage.	Slow for debugging and testing.
Execution time is more.	Execution time is less.
No security of source code.	Security of source code.
- Basic	- C, Cobol, Pascal, Fortran

Programming Concepts ?

- Computer requires instructions to be given for any job to be done.
- Students need to know as part of course and job.
- Later on just a matter of knowing the syntax.

Program Development Steps

In order to solve a problem using a computer it is necessary to evolve a detailed and precise step by step method of solution.

- Algorithm
- Flowchart
- Program

Algorithm

- Finite sequence of instructions (to solve a problem).
- The development of a proper procedure to get the required results.

Characteristics

- Inputs
- Precise & unambiguous processing rules
- Basic instructions
- Finite steps
- Outputs

Points for developing Algorithm

- Every procedure should carefully specify the input and output requirements.
- Meaning of variables should be clearly defined.
- The flow of program should generally be forward except for normal looping and unavoidable instance.

Example

Problem :

Obtain the percentage of marks obtained by a student in an examination.

Solution :

I/P : In the problem maximum marks and marks obtained is given.

O/P : The required result is percentage of marks and the formula used is

$$\% \text{ of marks} = \frac{\text{Marks obtained}}{\text{Maximum marks}} \times 100$$

Algorithm :

Step 1 : Read name, marks obtained, and maximum marks.

Step 2 : Divide marks obtained by maximum marks and store it in Per.

Step 3 : Multiply Per. by 100 to get percentage.

Step 4 : Write name and percentage.

Step 5 : Stop.

Flowchart

- A detailed graph which represents steps to be performed within the machine to produce the needed output.
- Algorithm represented in pictorial form.
- Requires only a few symbols in program charting to indicate the necessary operations.

Characteristics of FCs

- An aid in formulating and understanding algorithms.
- Easy visual recognition, a standard convention is used in drawing flow charts.
- Sequencing & repetition instructions easily visible.
- Helps in detecting, locating, and removing mistakes if the program fails to run to completion.
- The program FC acts as a guide or blueprint during the program preparation phase.

Characteristics of FCs….

- Leads to quicker grasp of relationships.
- It becomes a model of a program or system that can be broken down into detailed parts for study.
- Can be used as working models in the design of new programs and systems.
- Program Documentation.
- Aid in communicating the facts of a problem to those whose skills are needed in the solution.

Symbols used in Program FCs

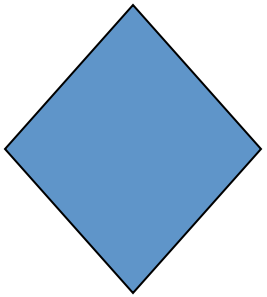
Symbols adopted by the American National Standards Institute (ANSI).

- Input/Output (Parallelogram)
  - Used to represent I/O operations.
  - Have two flow lines, entry & exit.

- Processing (Rectangle)
  - Storage & Arithmetic operations.
  - Have two flow lines, entry & exit.

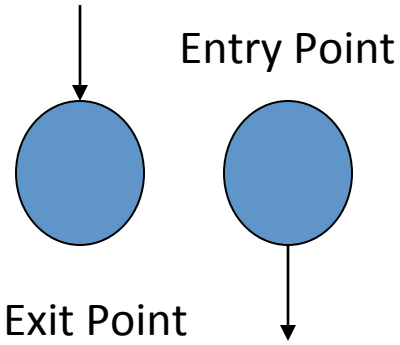
- Terminal (Rounded Rectangle)
  - Used to indicate START & STOP.
  - Has a single entry or exit line.

- Decision (Diamond)
  - Logic/comparison operations.
  - Has one entry and at least two exit paths or branches.
  - The exits are labeled with the answers to the decision question.



- Flow (Line with Arrow)
  - Most important in a Flow Chart.
  - Indicates the flow of logic of the program.
  - All the other flow chart symbols are connected by the flow line.

- Connector (Circle)
  - Used when additional flow lines might cause confusion and reduce understanding.
  - Two connectors with identical labels serve the same function as a long line.



- Predefined Process (Rectangle with end lines)
  - Certain processing operations are repeated in programs which are grouped into a separate procedure.
  - A single symbol replaces a number of operations that need not be detailed.

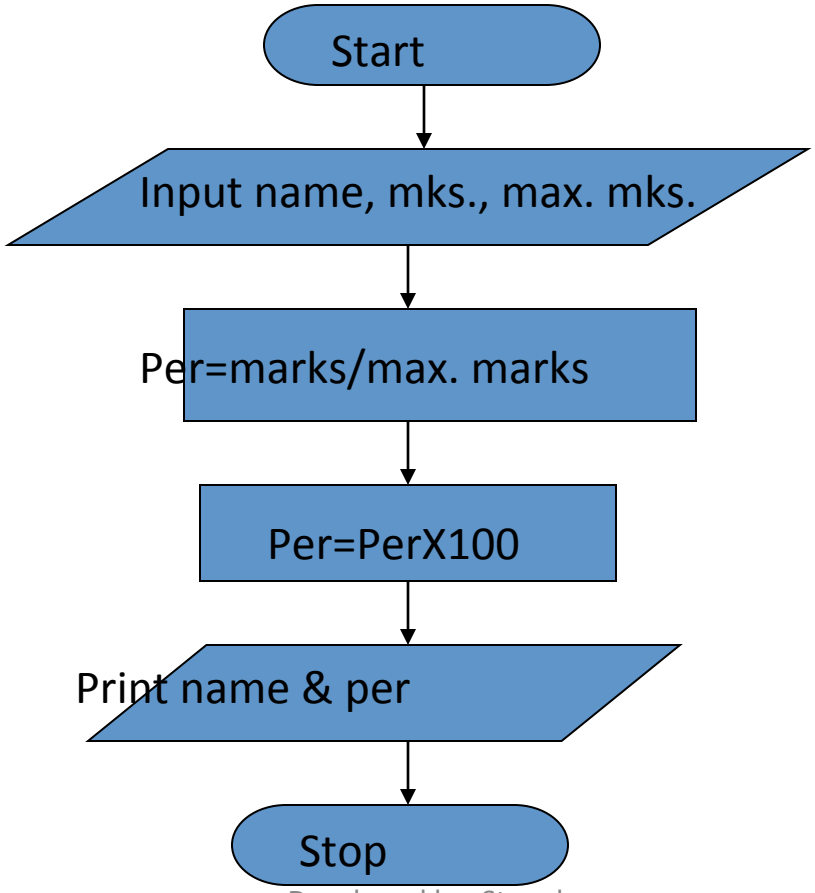
Limitations of FCs

- Complex and detailed charts are laborious to plan and draw.
- No standards determining the amount of detail that should be included in a chart.

Example

Problem :

Obtain the percentage of marks obtained by a student in an examination.



Program

- Expresses the flow chart/algorithm in a more precise and concise notation to be fed to the computer for execution.
- Machine-Independent : Primary objective is to facilitate a large number of people to use computers without the need to know in detail the internal structure of the computer.
- The specification of the sequence of computational steps in a particular programming language is termed as a program.

Programming Techniques

- **Linear Programming**

  When programming started the size and scope of the programs was small.

- **Structured Programming**

  Introduced as the size and scope of programs grew, the traditional linear approach to programming made programs unstructured and difficult to understand.

Linear Programming

- Straightforward programming in a sequential manner.
- Does not involve any decision making.

General model of a linear program :

1. Read a data value.
2. Compute an intermediate result.
3. Use the intermediate result to compute the desired answer.
4. Print the answer.
5. Stop.

Structured Programming

- It refers to the process in which we break the overall job down into separate piece of modules.
- Digital computers can make a decision, thus creating a branching point.
- If branching and looping can be use, then more complex iterative algorithms can be developed into complex programs.
- Complex programs that make them less error prone and easier to debug.

Choice of Modules

- Modules must be chosen in such a way that we can specify how they are to interact.
- There must be a contact between each modules.
- Contacts specify :
  - What the module will do ?
  - What inputs a particular module is to receive from the various other modules and what outputs it is required to provide for them ?

Advantages of Struct. Prog.

- Decreases the complexity of the program.
- Allows several programmers to code simultaneously.
- Allows reuse of common functions across programs.
- Isolates errors hence decreases debugging.
- Amendments to single module does not affect the rest of the program.
- As it is a standard method takes less time to write.
- Easy naming of modules help to locate easily in documentation.

Modular Design of Programs

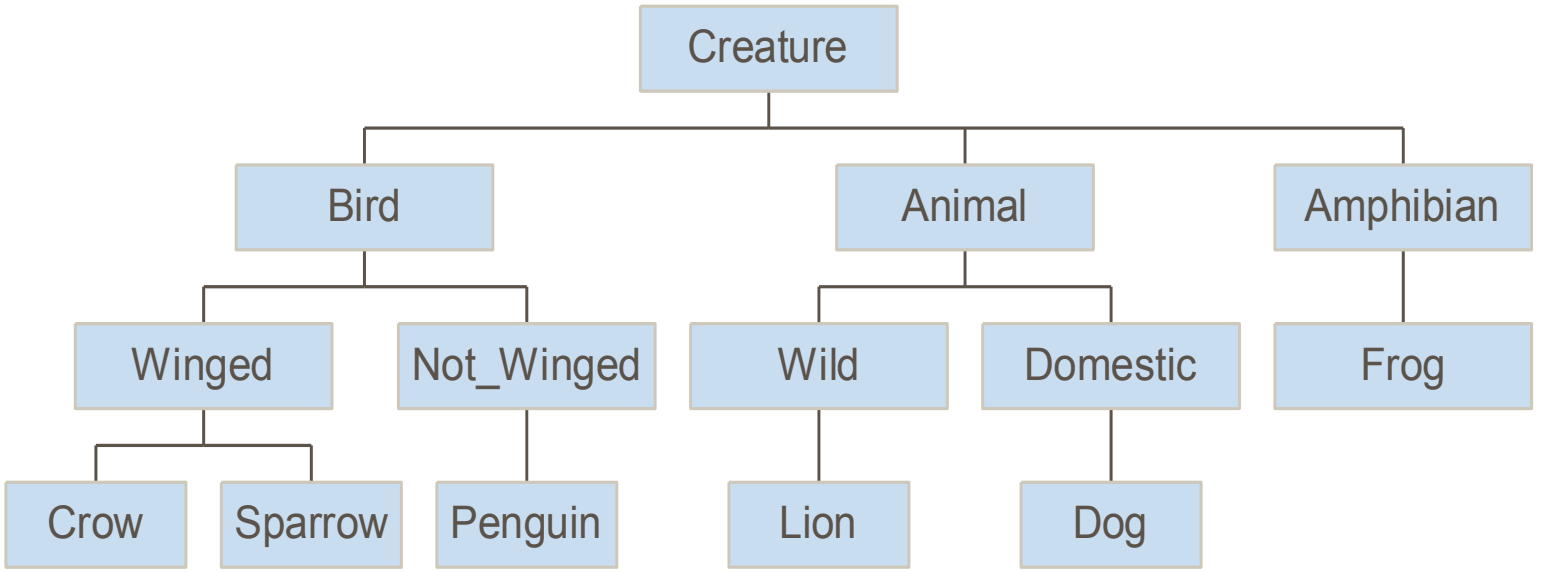
- Program is designed as a set of units referred to as blocks or modules.
- The modules reflect a logical flow for a computer program.
- Modules basically have :
  - Input
  - Output
  - Function
  - Mechanism
  - Internal Data
- Modules are arranged at different levels into a structured chart and all are connected.

Module Connection Rules

- Only one module at the top of the structure called the root module.
- The root passes control down the structure chart to the lower level modules. Control is always returned to the invoking module and a finished module should always terminate at the root.
- There cannot be more than one control relationship between any two modules on the structure chart. If module A invokes module B, then B cannot invoke module A.

Example

Structured Module Chart



Program Instruction Types

- Statements to establish the start of the program.
- Variable declaration.
- Program statements (blocks of code).
  - Expressions.
  - Programming Constructs.

Variable Declaration

- Place holders for data a program might use or manipulate.
- Variables are given names so that we can assign values to them and refer to them later to read the values.
- They are declared at the start because in order to use a variable within a program, the compiler needs to know in advance the type of data that will be stored in it.
- Variable typically stores *value* (content) of a given *type* (characteristic of the variable).

Variable Type

■ Integer	To store integer or "whole" numbers.
■ Real	To store real or fractional numbers (also called float to indicate a floating point number).
■ Character	A single character such as a letter of the alphabet or punctuation.
■ String	A collection of characters.

Expressions

- Expressions are made up of a combination of variables & operators which act on it.
- Operators work with respect to precedence & associativity rules set for the language.
- Some operators are :
  - Arithmetic (Mathematical)
  - Logical (Boolean)
  - Relational (Comparison)

Operators

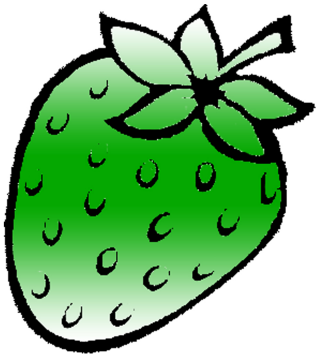
- Arithmetic
  - Add, Subtract, Multiply, Divide, Remainder.
- Logical
  - And, Or, Not.
- Relational
  - Less than, Greater than, Equal to, Not equal to, Less than equal to, Greater than equal to.

Programming Constructs

- Control statements "control" which sections of code in a program are to be executed.
- Types of Control Statements are :
  - Sequential - The default ordering of execution.
  - Selection (Conditional) - Controls which block of code within several alternatives is executed.
  - Iterative - controls how many times a block of code is executed.

Thank You

STRAWBERRY



/strawberrydevelopers

/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com