

### \* Structured programming.

Structured programming is the programming paradigm aimed at improving clarity, quality and development time of the computer program by making extensive use of subroutines, block structures, for and while loop. In contrast to using tests and jumps such as go to statement, structured programming is most frequently used with devices.

### \* Program Testing.

Program Testing is a process of running the program on sample data chosen to find errors if they are present. It is clear that the compiler can detect only syntax and semantic errors. All remaining errors can be detected at run time only. Therefore, it is necessary for the programmer to detect all such errors.

Basically there are two testing methods:

1. The Unit test method.
2. An Integration Test method.

1. The Unit test method  $\Rightarrow$  In this, program testing is done at subprogram level where a subprogram is designed to perform only one task.

2. An Integration test method  $\Rightarrow$  After testing of all subprograms, the data should be tested according to the specification of the complete program, without regard to initial details of the program. Such type of testing is made to find out the interfacing problems between different subprograms.

### \* Program Debugging :-

Program Debugging is the process of isolating and correcting the errors. Usually, a large program does not run correctly if it is executed for the whole for the first time. If it has errors, then it is necessary to determine where the errors are.

One simple and effective debugging tool is taking the snapshot of the program execution by inserting the "printf()" statement at key points in the main program. Once the errors are detected and corrected, these debugging (printf()) statements may be removed from the program.

\* Scaffolding :- We insert comments in the source code, so whenever we write program we place scaffolding into our program. It will be easy to delete once it is no longer needed.

\* Documentation :- Documentation is a written statement tests and comments that makes a program easier for others to understand, use and modify. For small programs, it is not necessary to have a detailed documentation because one can keep all details in one's mind and so needs documentation only to explain programs to someone else.

### \* Errors :-

When we code an algorithm it is not possible that our program is 100% error free. It means there are some errors present. So, it is programmer's duty to detect, isolate and correct any errors if they are present.

There are four types of errors:

1. Syntax error.
2. Run time error.
3. Logical error.
4. Latent error.

1. Syntax error  $\Rightarrow$  Each programming language has its own set of rules (such as "`<`") and the errors which violate such rules are referred as syntax errors.

Syntax errors are immediately detected and isolated during the compilation process.

Ex  $\rightarrow$  `int i=100, *radius;`

2. Runtime error  $\Rightarrow$  The errors which occur during the execution of program are referred as runtime errors. Runtime error comes when there is mismatch of basic datatype and make a reference of array element that falls ~~falls~~ outside its range.

Ex  $\rightarrow$  `int arr[100];`

3. Logical error  $\Rightarrow$  The errors which are related to the logic of the program are referred as logical error. Such errors occur due to wrong jumping, failure to satisfy a condition and any incorrect order of evaluation of statements.

Ex  $\rightarrow$  `total = a - b;`

4. Latent error  $\Rightarrow$  The errors which occur due to special value of data are referred as latent

error.  
Ex  $\rightarrow$  `y = 1/2`.

## # Difference between program testing and program Debugging.

### Program Testing

1. Testing is a process in which the program is validated.
2. Testing is a positive activity that checks the requirement of program specification.
3. Testing is said to be complete when all desired verifications among specifications are found valid.
4. Testing is a planned process in which a programmer has to make plan how and when to test in a program.
5. Testing can be performed in early stages as well as later stages of program development.

### Program Debugging

1. Debugging is a process in which errors are removed.
2. Debugging is a negative activity in the sense that it tracks unknown errors and removes them.
3. Debugging is finished when there is no error present.
4. Debugging is not a planned process actually it is stemmed from testing.
5. Debugging begins after the coding of algorithm because it requires an executable program.

\* Top-down approach.

→ Top-down design takes the whole software system as an entity and then decomposes it to achieve more than one sub-systems or components based on some characteristics.

→ Top-down design is more suitable when the software solution needs to be designed from scratch and specific details are unknown.

\* Bottom-up approach.

→ Bottom-up approach starts with the designing of most basic or primitive components and proceeds to higher-level components that use these lower-level components.

→ If a system is built from an existing software, then bottom-up approach is more suitable as it starts on some existing components.

## Unit 2

\* Data type.

Data type is a term, which is used to refer the kind of data that variables may hold in programming language. The general form of class of data items is known as data types.

Data types such as :

1. Unsigned char, Signed char, &
- 2) Int, long Int, Unsigned Int
- 3) double, long double, float.