Chapter-12
Debugging

Debugging:

Debugging is the process of detecting and removing of errors (also called as 'bugs') in a software code that can cause it to behave unexpectedly or crash. To prevent incorrect operation of a software or system, debugging is used to find and resolve bugs or defects. It gives step by step information about the execution of code to identify the fault in the program.

A debugger allows you, the programmer, to interact and inspect the running program, making it possible to trace the flow of execution and track down the problems.

Testing and Debugging

- Testing is a process of finding bugs or errors in a software product that is done manually by tester or can be automated. It is done by testers.

- Debugging is a process of fixing the bugs found in testing phase. Programmer or developer is responsible for debugging and it can't be automated. It is done by the developer or development team.

Difference between Testing and Debugging

Testing	Debugging
The purpose of testing is to find bugs and errors.	The purpose of debugging is to correct those bugs found during testing.
Testing is done by tester.	Debugging is done by programmer or developer.
It can be automated.	It can't be automated.
It can be done by outsider like client.	It must be done only by insider i.e. programmer.

Error:

Basically there are three types of errors in c programming:

1. Runtime Errors
2. Compile Errors
3. Logical Errors

1. Runtime Errors:

Runtime errors are those errors that occur during the execution of a c program and generally occur due to some illegal operation performed in the program.

Examples of some illegal operations that may produce runtime errors are:

- Dividing a number by zero
- Trying to open a file which is not created

2. Compile Errors:

Compile errors are those errors that occur at the time of compilation of the program. C compile errors may be further classified as:

- **Syntax Errors:** When the rules of the c programming language are not followed, the compiler will show syntax errors.

For example, consider the statement,

int a,b :

The above statement will produce syntax error as the statement is terminated with **:** rather than **;**

Most frequent syntax errors are:

a. Missing Parenthesis (**}**)
b. Printing the value of variable without declaring it
c. Missing semicolon (**;**)

- **Semantic errors :**

  This error occurs when the statements written in the program are not meaningful to the compiler.

For example, consider the statement,

b+c=a;

In the above statement we are trying to assign value of a in the value obtained by summation of b and c which has no meaning in c. The correct statement will be

a=b+c;

3.Logical Errors:

On compilation and execution of a program, desired output is not obtained when certain input values are given. These types of errors which provide incorrect output but appears to be error free are called logical errors.

- for(i = 0; i < 3; i++)**;** // logical error : a semicolon after loop

- if(a=1) // logical error: assignment operator ”=” is used instead of “==” Comparison operator

***