

## UNIT IV

### Introduction:

A Software Application needs to be error-free before going out in the market. Customer satisfaction is of utmost importance for any organization and only a bug-free product can keep your customer happy.

### What is Debugging?

In the context of software engineering, debugging is the process of fixing a bug in the software. In other words, it refers to identifying, analyzing and removing errors. This activity begins after the software fails to execute properly and concludes by solving the problem and successfully testing the software. It is considered to be an extremely complex and tedious task because errors need to be resolved at all stages of debugging.

### Why do we need Debugging?

The process of debugging begins as soon as the **code** of the **software** is written. Then, it continues in successive stages as code is combined with other units of programming to form a software product. Debugging has many benefits such as:

- It **reports an error condition immediately**. This allows earlier detection of an error and makes the process of software development stress-free and unproblematic.
- It also provides **maximum useful information** of data structures and allows **easy interpretation**.
- Debugging assists the developer in **reducing useless and distracting information**.
- Through debugging the developer can **avoid complex one-use testing code** to save time and energy in software development.

**Debugging Process : Steps involved in debugging are :** The different steps involved in the process of debugging are



- **1. Identify the Error:** A bad identification of an error can lead to wasted developing time. It is usual that production errors reported by users are hard to interpret and sometimes the information we receive is misleading. It is import to identify the actual error.
- **2. Find the Error Location:** After identifying the error correctly, you need to go through the code to find the exact spot where the error is located. In this stage, you need to focus on finding the error instead of understanding it.
- **3. Analyze the Error:** In the third step, you need to use a bottom-up approach from the error location and analyze the code. This helps you in understanding the error. Analyzing a bug has two main goals, such as checking around the error for other errors to be found, and to make sure about the risks of entering any collateral damage in the fix.
- **4. Prove the Analysis:** Once you are done analyzing the original bug, you need to find a few more errors that may appear on the application. This step is about writing automated tests for these areas with the help of a test framework.
- **5. Cover Lateral Damage:** In this stage, you need to create or gather all the unit tests for the code where you are going to make changes. Now, if you run these unit tests, they all should pass.
- **6. Fix & Validate:** The final stage is the fix all the errors and run all the test scripts to check if they all pass.