LECTURE 11: PROGRAM DOCUMENTATION

Contents

10.1 Introduction 1
10.2 Advantages of Documentation 1
10.3 Example Documents 2
10.4 When should program documentation be done? 2
10.5 What information should be in the program documentation? 3
10.6 How should program documentation be done? 3

10.1 Introduction

Any written text, illustrations or video that describe a software or program to its users is called **program or software document**. User can be anyone from a programmer, system analyst and administrator to end user. At various stages of development multiple documents may be created for different users. In fact, **software documentation** is a critical process in the overall software development process.

In modular programming documentation becomes even more important because different modules of the software are developed by different teams. If anyone other than the development team wants to or needs to understand a module, good and detailed documentation will make the task easier.

These are some guidelines for creating the documents −

- Documentation should be from the point of view of the reader
- Document should be unambiguous
- There should be no repetition
- Industry standards should be used
- Documents should always be updated
- Any outdated document should be phased out after due recording of the phase out

10.2 Advantages of Documentation

These are some of the advantages of providing program documentation −

- Keeps track of all parts of a software or program
- Maintenance is easier
- Programmers other than the developer can understand all aspects of software
- Improves overall quality of the software
- Assists in user training

- Ensures knowledge de-centralization, cutting costs and effort if people leave the system abruptly

10.3 Example Documents

A software can have many types of documents associated with it. Some of the important ones include −

- **User manual** − It describes instructions and procedures for end users to use the different features of the software.
- **Operational manual** − It lists and describes all the operations being carried out and their inter-dependencies.
- **Design Document** − It gives an overview of the software and describes design elements in detail. It documents details like **data flow diagrams, entity relationship diagrams**, etc.
- **Requirements Document** − It has a list of all the requirements of the system as well as an analysis of viability of the requirements. It can have user cases, reallife scenarios, etc.
- **Technical Documentation** − It is a documentation of actual programming components like algorithms, flowcharts, program codes, functional modules, etc.
- **Testing Document** − It records test plan, test cases, validation plan, verification plan, test results, etc. Testing is one phase of software development that needs intensive documentation.
- **List of Known Bugs** − Every software has bugs or errors that cannot be removed because either they were discovered very late or are harmless or will take more effort and time than necessary to rectify. These bugs are listed with program documentation so that they may be removed at a later date. Also they help the users, implementers and maintenance people if the bug is activated.

10.4 When should program documentation be done?

When designing your program, you must spend time thinking about how to structure your program, what modules are needed, and the algorithms and processes you will use in the modules. You must think about what sort of data structures and objects (e.g., arrays, files or linked lists) are needed. This thinking must be done before you start coding, or you will find yourself wasting time writing useless code that is full of errors. It is very important to record this creative process so that the programmers that follow you do not duplicate work that you have already done.

Before writing the code, you should write the documentation to describe the design of each component of your program. Writing documentation for the modules before writing the code helps you define exactly what each module should do and how it will interact with other modules. Focusing on the design and the steps needed to solve the problem can help prevent errors in the completed program.

10.5 What information should be in the program documentation?

For an individual module, it is important to record

1) who has written the module,
2) when the module was written or modified,
3) why the module was written or modified,
4) how the module interacts with other modules,
5) what special algorithms were used, if any, and
6) acknowledge outside sources for ideas and algorithms.

For data structures, it is important to record

1) what data structure is used,
2) why a particular structure was used,
3) what data is contained in the structure, and
4) how the data structure is implemented.

10.6 How should program documentation be done?

The following sections contain guidelines for documentation that should appear in a C++ program file containing a main program and some functions. Documentation or comment lines begin with a double slash (//) and all text to the end of the line is considered a comment. Documentation should appear

• before the main program (program header)
• before each function (function header)
• before or next to lines of code within a function (line comments)