Lesson 1: Introduction

1.1. Introduction

In this lesson, you will learn about software crisis, software quality issues, procedure oriented and object oriented programming.

1.2. Lesson objectives

By the end of this lesson, you will be able to:

- Describe software crisis
- Describe various programming paradigms
- Define common terminologies in object oriented programming

1.3. Lesson outline

This lesson is structured as follows:

1.1. Introduction
1.2. Lesson objectives
1.3. Lesson outline
1.4. Software crisis
1.5. Software quality
1.6. Procedure oriented and object oriented programming paradigms
1.7. Object oriented programming concepts
1.8. Advantages of object oriented programming
1.9. Disadvantages of object oriented programming
1.10. Revision
1.11. Summary
1.12. Suggested reading

1.4. Software crisis

Developments in software technology continues to be dynamic. This is because new tools and techniques are announced in quick succession. Consequently, software engineers and industry are continuously looking for new approaches to software design and developments. These rapid advances have created a situation of crisis in the software industry. The critical issues are:

- How to represent real life entities of the problem in the system
- How to design systems with open interfaces
- How to ensure reusability and extensibility of modules
- How to develop modules that are resilient to any changes in future
- How to improve software productivity and decrease cost
- How to improve software quality

- How to manage time schedules

1.5. Software quality

Software quality may refer to how well the software complies with or conforms to a given design, based on functional requirements or specifications as well as non-functional requirements. The quality issues include:

- Correctness: No errors
- Maintainability-can be modified
- Reusability – Need to rewrite modules
- Openness and interoperability-conformity
- Portability –run on different platforms
- Security- Data and code protection
- Integrity- consistency in performance and computations
- User friendliness- easy to interact with

1.6. Procedure oriented and object oriented programming paradigms

[a]. Procedure oriented programming (POP) paradigms

In this programming paradigm, the problem is viewed as a sequence of things to be done (steps). A number of functions are written to accomplish these tasks. The characteristics of POP are:

- Emphasis is doing things (procedure)
- Large programs are divided into smaller programs (functions)
- Most functions share global data
- Data can move openly around the system from one function to another The function transforms data from one form to another.
- This paradigm employs top-down approach for program design.

[b]. Object oriented programming(OOP) paradigms

In this programming paradigm, the problem is decomposed into a number of entities called objects .Data and functions are then built around these objects. The characteristics of OOP are:

- Emphasis is on data rather than procedure
- Programs are divided into objects
- Data structures are designed to characterize the objects
- Functions that operate on these data are tied together with data structures
- Data is hidden and cannot be accessed by external functions
- The objects can communicate with each other through functions
- New data and functions can be added when necessary The paradigm employs bottom up approach in program design.

1.7. Object oriented programming concepts:

Some of the basic concepts:

Object: This is basic run-time entities in an object oriented system. Also referred to as instance of a class.

Class: Refer to a collection of objects of similar type. Also defined as a template for creating objects.

Data abstraction: Refers to a representation of essential details about attributes without including background details.

Functional abstraction: Refer to a representation of essential details about a function without including background details (how the function is implemented).

Encapsulation: Refer to the process of wrapping up of data and functions into a single unit.

Information hiding: Refer to the insulation of data from direct access by any program.

Inheritance: The process by which objects of one class acquire properties of objects of another class.

Polymorphism: The ability of operation to exhibit different behaviors in different instances e.g. operator overloading and function overloading.

Dynamic binding: Also referred to as late binding. This is where the code associated with a given procedure call is not known until the time of the call at run time. It is mainly used in polymorphism and inheritance.

Message passing: Refer to specifying the name of the object, name of the function (message) and information to be send. Example **rect1.area (2,5).**

1.8. Advantages of object oriented programming

- Improved software development productivity: This is because Object-oriented programming is modular, as it provides separation of duties in object-based program development.
- Faster development: Reuse enables faster development.
- Lower cost of development: The reuse of software also lowers the cost of development.
- Higher-quality software: Faster development of software and lower cost of development allows more time and resources to be used in the verification of the software.

1.9. Disadvantages of object oriented programming

- **Steep learning curve:** The thought process involved in object-oriented programming may not be natural for some people, and it can take time to get used to it.
- **Larger program size:** Object-oriented programs typically involve more lines of code than procedural programs.

- **Slower programs:** Object-oriented programs are typically slower than procedure oriented programs, as they typically require more instructions to be executed.

1.10. Revision questions

[a]. Define object oriented programming.

[b]. Discuss any four applications of object oriented programming [c]. Differentiate between:

- Object and class
- Data abstraction and data encapsulation Inheritance and polymorphism.

1.11. Summary

In this lesson, you have learnt about software crisis, software quality issues. You have also looked at procedure oriented and object oriented programming paradigms.

1.12. Suggested reading

[1]. Object oriented programming with C++ by E Balagurusamy 3rd ed; publisher: Tata Mcraw Hill

[2]. Sams teach yourself c++ in 24 hours by Jesse Liberty and Rogers Cadenhead. [3]. Object oriented programming in c++ by Joyce Farrel [4]. Object-oriented programming with c++ by Sourav Sahay.