Programming Refresher

Programming Basics

Learning Objectives

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

- Explain software and the need of software
- Categorize different types of software
- Explain distribution of software
- Illustrate procedural and object-oriented programming
- Summarize program structure with its syntax

Business Scenario

ABC is a supply-chain management company. The development team at ABC is working on developing software to manage the supply chain of its client's products. This can be a web, or a desktop application installed at all warehouses.

The company must determine what type of software will best meet the given demand and make use of programming procedures to implement it.

To address these challenges, we will explore the following:
- What is software?
- Types of software
- Distribution of software
- Implementation of procedural and object-oriented programming

Introduction to Software

Discussion

Software

Why is software necessary? How do they make our lives easy?

- What is software?
- Why do we need software?

What Is Software ?

- Software is a set of instructions written for the computer to understand, interpret, and execute.
- Software programs are structured, organized, and formally written in a computer language. Hardware is used to run software programs.
- Without software support, hardware cannot increase its capabilities. The network and computer systems cannot function without software programs.

Software Examples

A few examples of software are:

Operating system

Internet browser

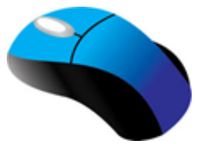
Adobe

Spreadsheet

Office

Hardware Examples

Hardware is a physical electronic device used in or with a machine. Some examples of hardware in a computer are:



Mouse

Keyboard

Ethernet cords

Hard drive

Why Do We Need Software?

1. Software makes human tasks easier and provides a consistent way of executing instructions with minimal to zero error, unlike execution done by human beings.

2. Software can be customized for any organization or user and designed for the mass market, such as commercial off-the-shelf software.

3. For instance, a calculator comes with software that makes regular calculations simpler and faster.

Why Do We Need Software?

- Software has evolved with time with the advancement in technology.
- In today's era, software can be used to create complex enterprise applications for various industries, such as transportation, medicine, defense, telecommunications, and others.

Software

Why is software necessary? How do they make our lives easy?

- What is software?

Answer: Software is a set of instructions written for the computer to understand, interpret, and execute.

- Why do we need software?

Answer: Software makes human tasks easier and provides a consistent way of executing instructions with minimal to zero error.

Categories of Software

Discussion

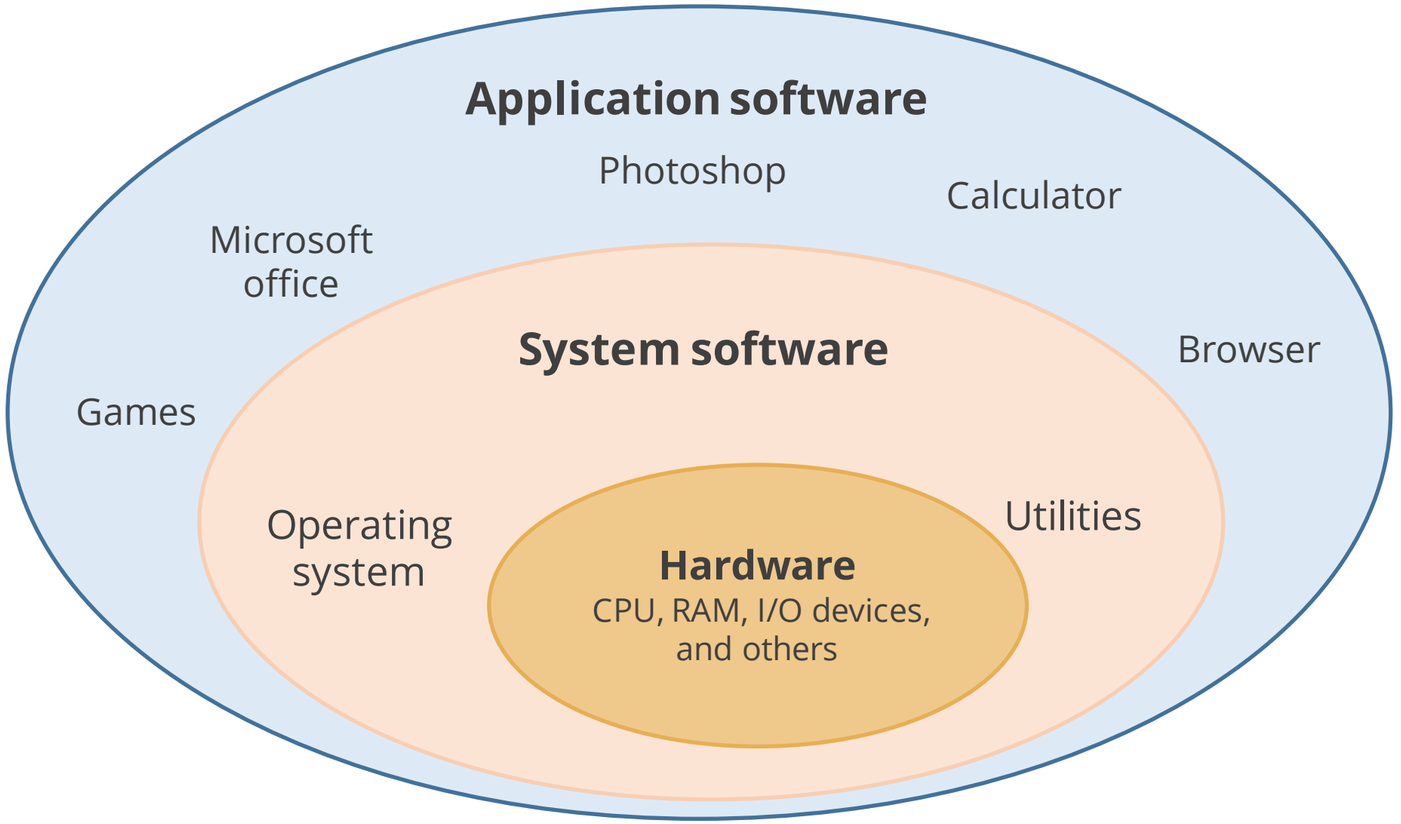
Types of Software

What are the different types of software?

What type of software is Excel?
What type of software is Windows?

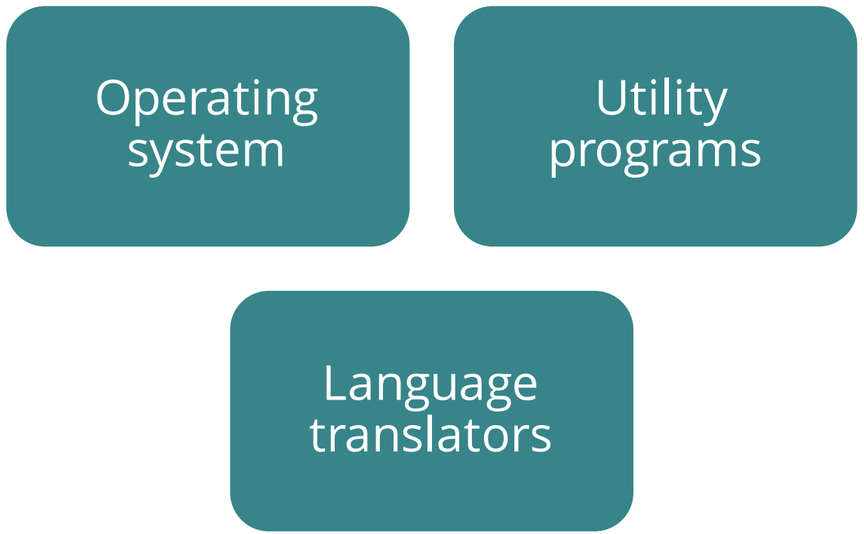
Types of Software

Software can be broadly classified in two categories:



- System software
- Application software

System Software



- It acts as an interface between hardware and application software.
- These software run background processes.
- Machine understands binary language and system software helps to convert human instructions into machine understandable instructions.
- System software can be categorized based on its functions.

Application Software

It is a set of programs that are used to perform a specific task on a system. These are customized to solve specific types of problems and provide functionalities.

Example

- Excel is used for operations with numbers.
- PowerPoint is used to create presentations.
- A browser is used for surfing the internet.
- A library management system provides functionalities related to the library.
- Tally is used for accounting.

Distribution of Software

The software can be distributed in many ways :

Freeware

It refers to software that is freely available without any charge. It is typically released under a freeware license, which allows users to use, copy, distribute, and modify the software without any restrictions.

Shareware

It is typically released under a shareware license. It is time-limited free software that requires payment for continued use after the trial period.

Distribution of Software

The software can be distributed in many ways :

Open-source software

It is often released under an open-source license, and enables unrestricted usage, modification, and distribution. Its source code is generally accessible to the public, fostering inspection and improvement by a wide range of individuals.

Licensed software

It is protected by copyright and must be purchased from a vendor. It is governed by a proprietary license, granting users usage rights but not permitted to modify or distribute the software.

Types of Software

What are the different types of software?
Answer: Software is broadly classified into system software and application software

- What type of software is Excel?

Answer: Excel is an application software as it is a set of programs used to perform a specific task on a system.

- What type of software is Windows?

Answer: Windows is an operating system (OS) software. It is a widely used OS developed by Microsoft that provides a platform for running applications and managing computer hardware.

Programming Models

Discussion

Procedural Programming

Scenario: You are a data scientist in your organization and have been put into a project that deals with critical information. While coding for an application, you hide everything but the necessary information about an item to simplify it and boost productivity.

In simple terms, you decide to showcase only the data that is relevant and hide all the other critical and unnecessary data from the user.

Which OOPs concept is being used here?

Procedural Programming

- Procedural programming is a programming paradigm built around the idea that programs are sequences of instructions to be executed.
- A program is divided into multiple sections called modules, subroutines, or block structures, and the aim is to improve the clarity, quality, and development time of a computer program.
- Instead of using a go-to statement like assembly language, procedural programming (structured programming) uses blocks, control structures, and loops.
- Example: Pascal, C

Object-Oriented Programming Language

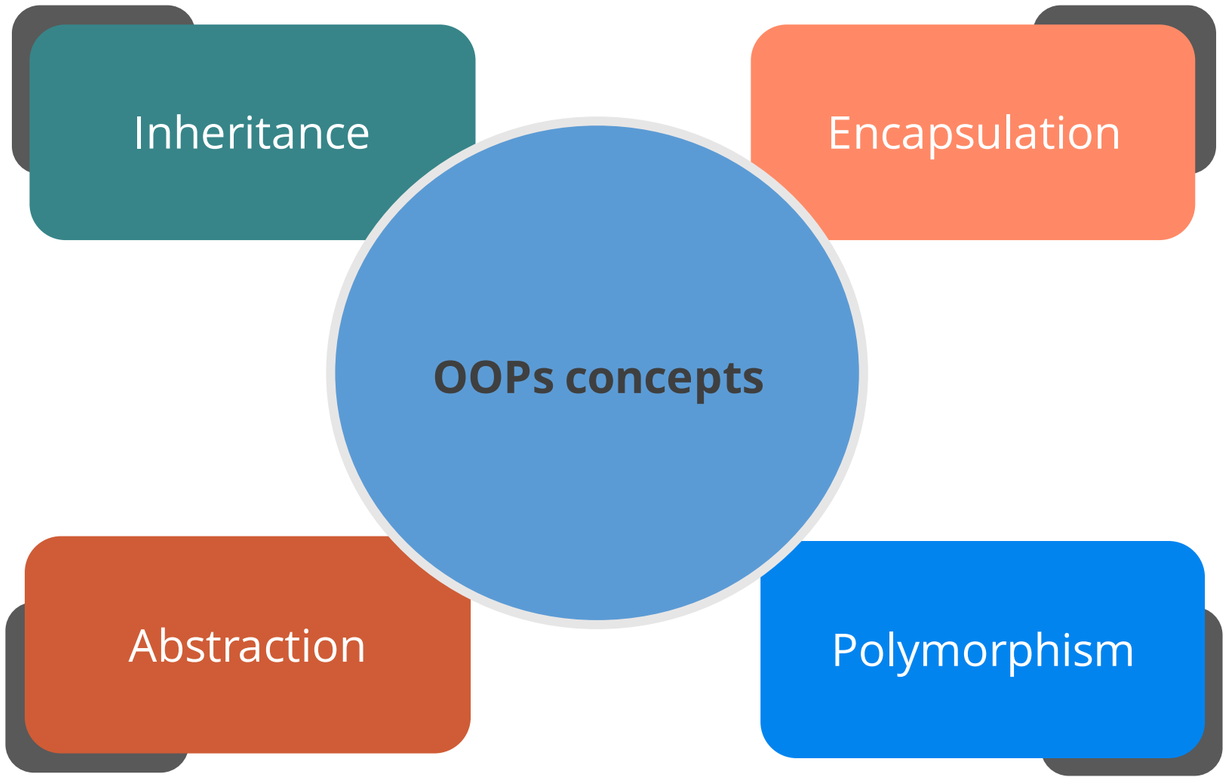
OOPs refer to languages that use objects in programming. It aims to implement real-world entities such as inheritance, information hiding, and polymorphism in programming.

Advantages of OOPs

- OOPs are faster and easier to execute.
- OOPs provide a clear structure for the programs.
- OOPs are more secure than procedural languages.
- Objects can move freely within member functions.
- OOPs help in creating reusable applications with less code and shorter development time.

OOPs: Concepts

OOPs focus on the following principles:



Procedural Programming

Scenario: You are a data scientist in your organization and have been put into a project that deals with critical information. While coding for an application, you hide everything but the necessary information about an item to simplify it and boost productivity.

In simple terms, you decide to showcase only the data that is relevant and hide all the other critical and unnecessary data from the user.

Which OOPs concept is being used here?

Answer: The OOPs concept being used here is data abstraction.

Program Structure

Discussion

Program Structure

- How do you decide if a programming language is a compiler-based language or an interpreter-based language?
- What is the difference between a compiler-based language and an interpreter-based language?

Program Structure

1. A program consists of multiple components that interact with each other and have interrelationships. A well-structured program might be complex but not complicated.

2. In a well-structured program, the division into components follows a recognized principle, such as information hiding, and the interfaces between components are explicit and simple.

Program Structure

3

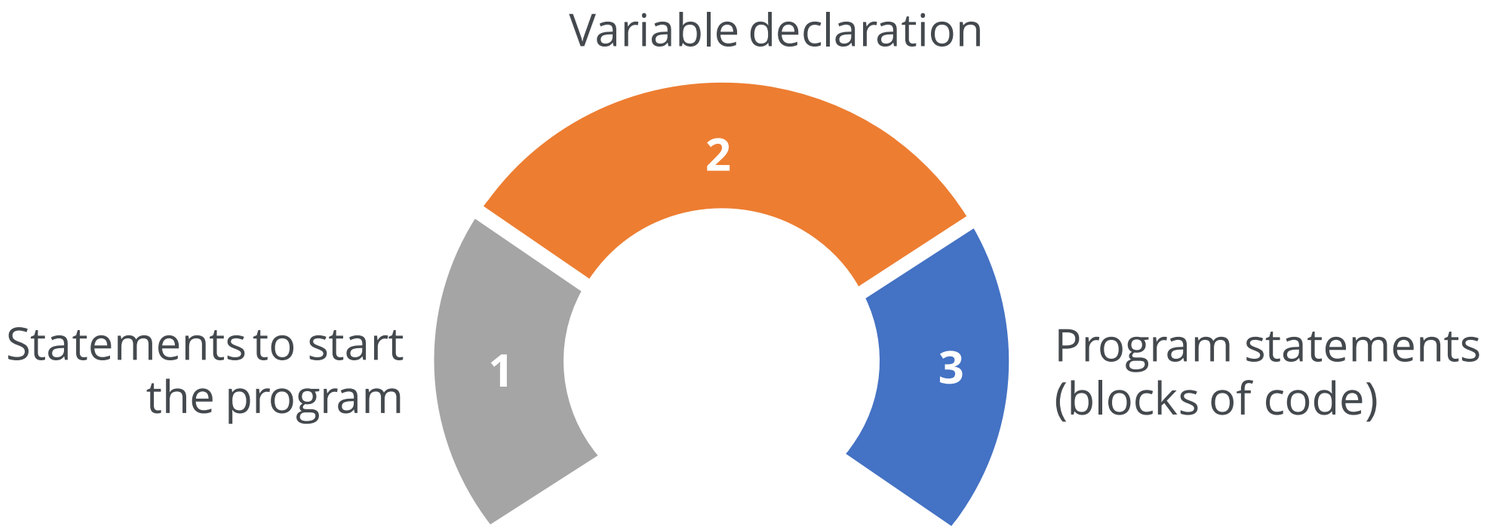
In contrast, a poorly structured program has interfaces that are implicit and difficult to understand, and the division into components is largely arbitrary (or non-existent.)

4

A well-structured program employs appropriate data structures and program units with a single-entry point and a single exit point, while a poorly structured program employs arbitrary data structures and flow of control.

Structured Program

Virtually all structured programs share a similar overall pattern:



Syntax in Different Languages

The programming languages listed below support the following syntax:

Language	Syntax example
C	#include<stdio.h> void main(){printf(“Hey Hi!!”);}
C++	#include<iostream> int main(){cout<<(“Hey Hi!!”; return 0;}
Pascal	program HelloWorld; begin writeln(“Hey Hi!!”); end
Oracle PL/SQL	CREATE OR REPLACEPROCEDURE helloworld AS BEGIN DBMS_OUTPUT.PUT_LINE(‘HELLO WORLD’); END;

Syntax in Different Languages

The programming languages listed below support the following syntax:

Language	Syntax example

Java	class helloworld{
	public static void main (String args []) {
	System.out.printIn(“Hello World!!”);}}

Perl	#!/usr/bin/env perl use strict; use warnings; print “Hello World”;
Basic	print “Hello World”

Types of Programming Language

Programming languages can be compiled or interpreted.

Compiler-based language	Interpreter-based language
A compiler takes the entire program in one go.	An interpreter takes a single line of code at a time.
The compiler generates an intermediate machine code.	The interpreter never produces any intermediate machine code.
The compiler is best suited for the production environment.	An interpreter is best suited for a software development environment.
Programming Languages, such as C, C++, C#, Scala, and Java are compiler-based languages.	Python, Ruby, PHP, and Perl are interpreter-based languages.

Program Structure

- How do you decide if a programming language is a compiler-based language or an interpreter-based language?

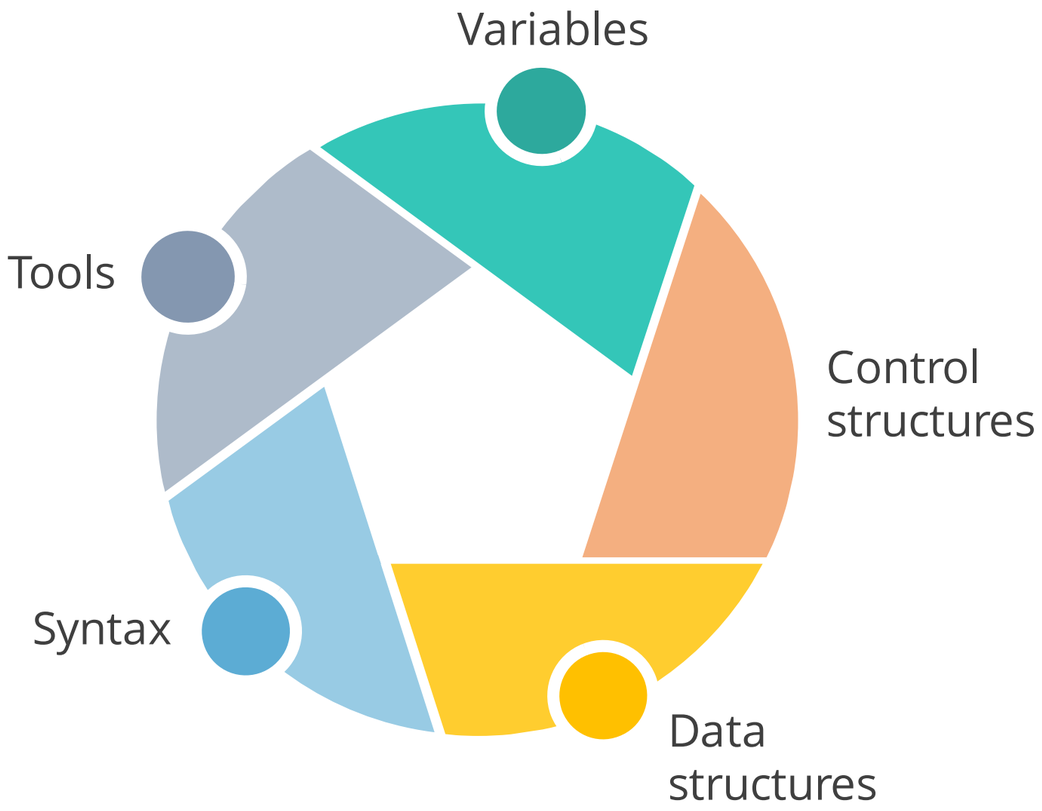
Answer: If a language is being used for the production environment, then it is a compiler-based language. If a language is being used for the software development environment, then it is an interpreter-based language.

- What is the difference between a compiler-based language and an interpreter-based language?

Answer: One of the difference is that a compiler-based language takes the entire program in one go, whereas an interpreter-based language only takes a single line of code at a time.

Programming Concepts

There are five basic concepts used in any programming language.



Variables

A variable is a storage location and an associated symbolic name that contains some known or unknown quantity or information, that is, a value.

Example

- int no;
- string name;
- double amount;

In some programming languages, it is important to specify the type of value that will be stored in the variable, which is known as the data type.

Control Structures

Control structures direct the program's flow depending on a decision, causing it to jump from one line to another or return a section of code. It can be done using an if-else construct.

Example

```
if (age >= 18)
“a person can vote “
else
“a person cannot vote “
```

It controls the flow of the program structure. When a program runs, a compiler reads and executes the user program line by line. This is called the top to bottom approach.

Data Structures

Data structures are a way of storing and organizing data in a computer in a way that can be effectively used. Lists, dictionaries, and arrays are a few examples of data structures.

Example

To store a list of contacts for 100 people, instead of creating 100 variables, a data structure like a list or array can be used to store these contacts.

Syntax

Syntax defines a set of rules that specify the combinations of symbols that are correctly structured statements or expressions in that language.

Example

```
String name = “Jack Ford”;
```

Symbols, characters, and words in a programming language have special meanings and rules to enable the compiler to understand the instructions.

Tools

Tools in programming are the editors that help the user code quickly and efficiently.

Example

- Eclipse
- Atom
- Jupyter
- Microsoft Visual Studio

There are various IDEs (Integrated development environments) available for coding based on the technology and language.

Key Takeaways

- Software is a set of instructions given to the computer to process and give the desired output.
- Software can be categorized based on the purpose into system or application software.
- Procedural and object-oriented programming are the two types of programming models.
- A program needs to be well-defined and structured to be able to enhance, scale, and extend its capabilities for future requirements.

Knowledge Check

Knowledge Check 1

The language understood by a computer without translation is called _____

A. Command language

B. High-level language

C. Assembly language

D. Machine language

Knowledge Check

1

The language understood by a computer without translation is called _____

A. Command language

B. High-level language

C. Assembly language

D. Machine language

The correct answer is **D**

The computer understands machine language without translation.

Knowledge Check 2

Which of the following software defines an operating system and utility program?

A. System software

B. Application software

C. Device driver

D. Customized software

Knowledge Check 2

Which of the following software defines an operating system and utility program?

A. System software

B. Application software

C. Device driver

D. Customized software

The correct answer is **A**

Operating systems and utility programs are examples of system software.

Knowledge Check

3

Which feature of OOPs indicates code reusability?

A. Inheritance

B. Abstraction

C. Polymorphism

D. Encapsulation

Knowledge Check

3

Which feature of OOPs indicates code reusability?

A. Inheritance

B. Abstraction

C. Polymorphism

D. Encapsulation

The correct answer is **A**

Inheritance is used for code reusability.