Dr.SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE
(Autonomous)
Accredited by NAAC – UGC with 'A+ Grade (Cycle IV)
(Recognised by UGC, Approved by AICTE & Affiliated to Bharathiar University)
Coimbatore- 49

DEPARTMENT OF COMMERCE WITH INFORMATION TECHNOLOGY

21UCI507 -Business Information Technology

Programming Languages

Mrs.M.Viveka, MCA., M.Phil., (Ph.D).,

Assistant Professor,

Department of Commerce with Information Technology

Programming Language

- Formal language that is used to write instructions for a computer to perform specific tasks.
- It helps humans communicate with computers through code.
- "A programming language is a set of rules and symbols used to write programs that instruct a computer to perform operations.“
- It converts human logic into machine-understandable commands.

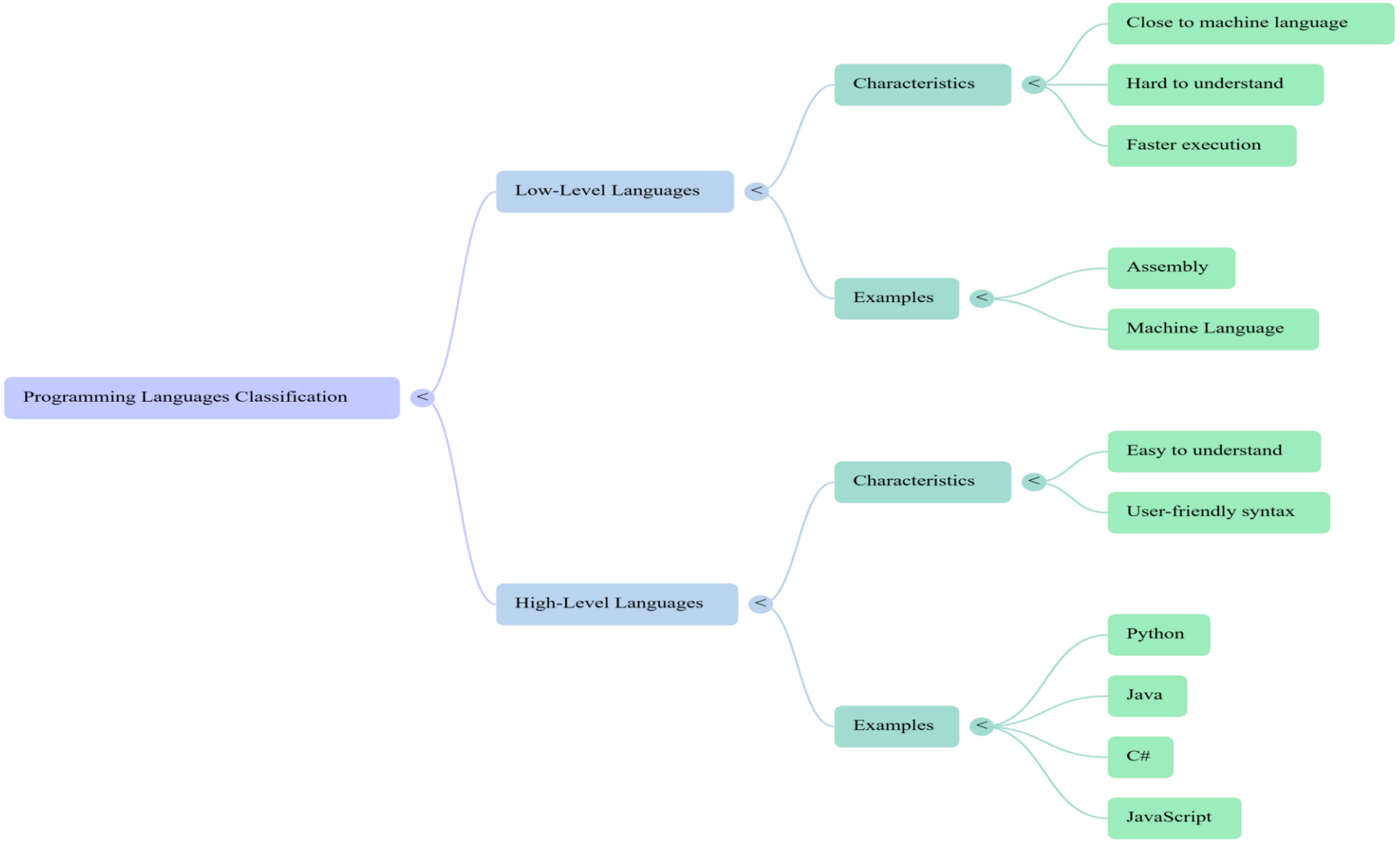
Examples:

- Python, C, C++, Java

Need for Programming Languages

- To develop software and applications
- To solve complex problems
- To control hardware devices
- To automate tasks
- To process data efficiently

Types of Programming Languages



Levels of Programming Languages

1.Machine Language

- Binary format (0s and 1s)

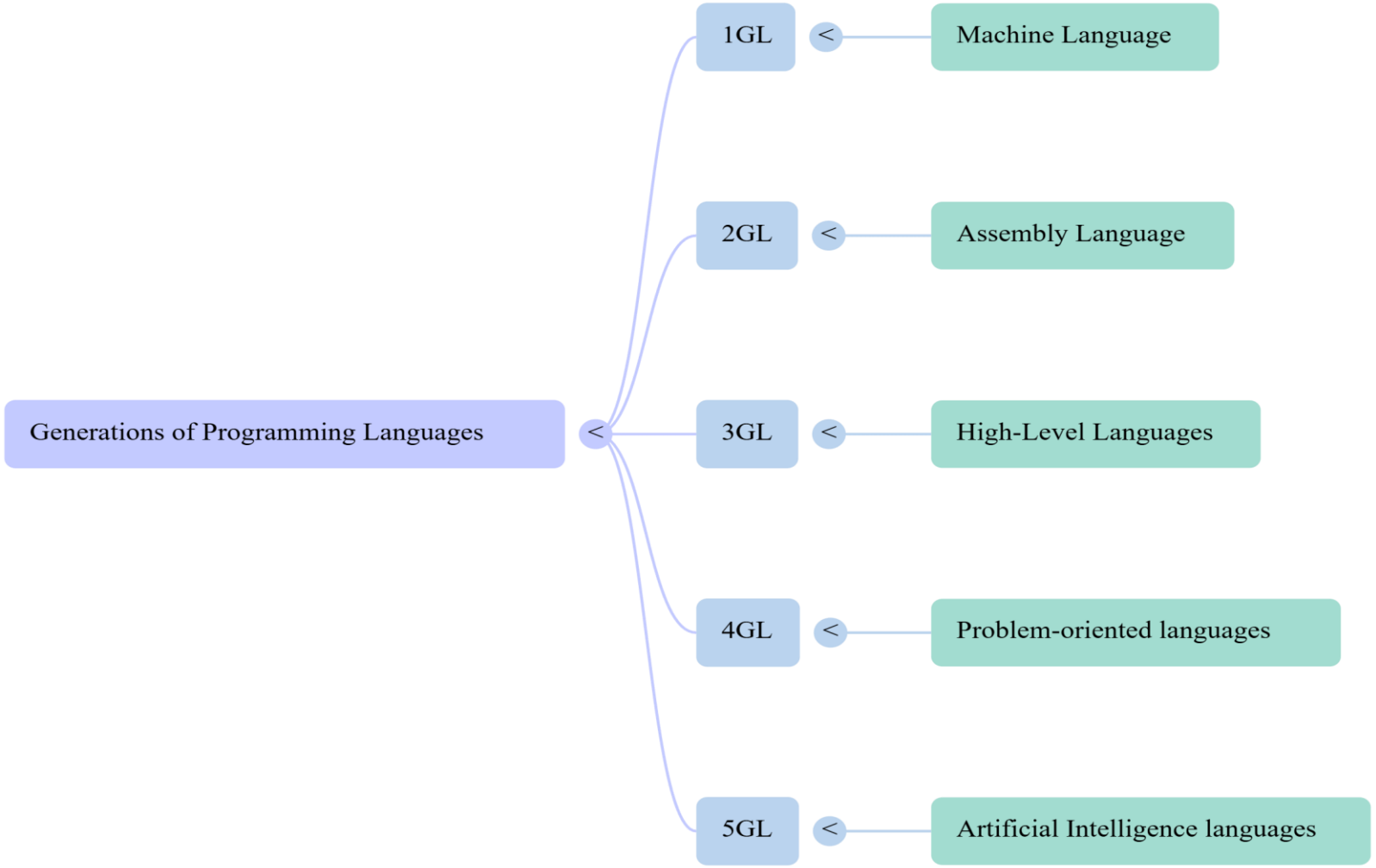
2.Assembly Language

- Uses mnemonics

3.High-Level Language

- English-like syntax

Generations of Programming Language



Language Translators

- Used to convert high-level code into machine code
  - ❑ Compiler
  - ❑ Interpreter
  - ❑ Assembler

Example:

- ➢ Python uses Interpreter
- ➢ C uses Compiler

Popular Programming Languages

- Python – General purpose, AI, Web
- Java – Application development
- C – System programming
- C++ – Game development
- JavaScript – Web development

Characteristics of Programming Languages

- ✓ Simplicity
- ✓ Readability
- ✓ Portability
- ✓ Efficiency
- ✓ Flexibility

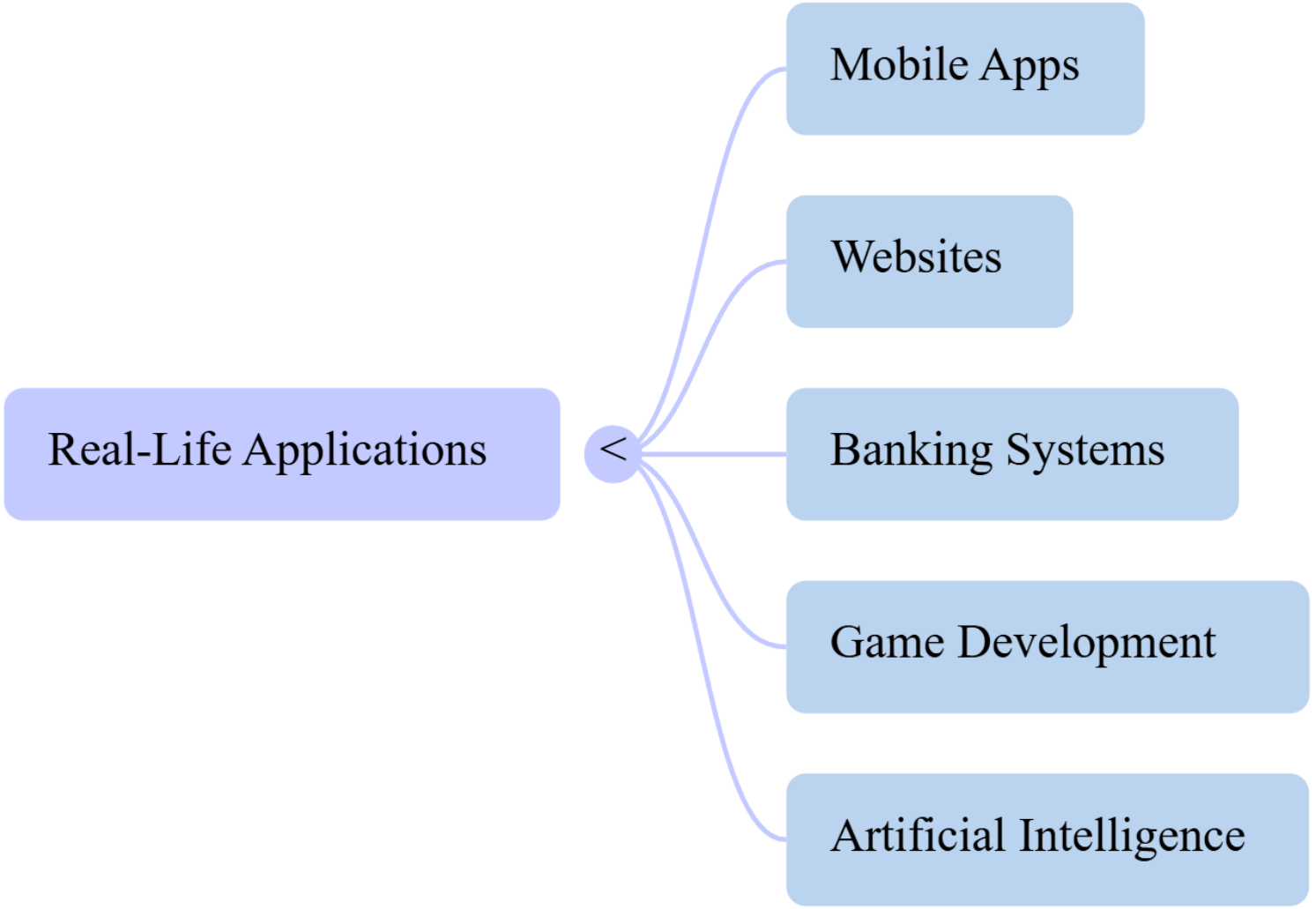
Advantages of Programming Languages

- ✓ Helps create dynamic applications
- ✓ Improves automation
- ✓ Solves real-world problems
- ✓ Increases productivity
- ✓ Enables technological

Programming Languages vs Markup Languages

Programming Language	Markup Language
Has logic & conditions	Only structure
Example: Python	Example: HTML

Real-Life Applications



Case Study

Choosing the Best Programming Language for a College Mini Project Portal

1. Empathize (Understand Users & Their Needs)

The college wants to create a **Mini Project Submission Portal** where:

•Students upload project files

•Faculty review and grade

•Departments maintain records

•which programming language should be used to build this system.

Users Identified:

UG Students, Faculty Members, Department Technical Team

User Needs:

- It should support **web development**.
- Should allow **fast development** of the portal.
- Should work on **any operating system**
- Should have **large community support** and tutorials.

Case Study

2. Define (State the Problem Clearly)

Students face difficulty choosing the correct programming language for their portal project.

Problem Statement:

"How might we select a programming language that is simple, flexible, and suitable for building a student mini-project portal?"

3. Ideate (Brainstorm Possible Languages)

The team considers multiple languages:

Options:

- **C** – Fast, but difficult for web development
- **Java** – Powerful but heavy and complex for beginners
- **Python** – Easy, flexible, many libraries
- **JavaScript** – Excellent for web development
- **PHP** – Simple and commonly used for portals
- **C#** – Good but requires Windows-based environment

Case Study

Comparison Summary:

Language	Pros	Cons
C	Efficient	Not ideal for web apps
Java	Secure	Complex syntax
Python	Easy, fast, many libraries	Slightly slower
JavaScript	Web-friendly	Needs frameworks
PHP	Simple for websites	Limited modern use
C#	Good for enterprise	Windows-based

Case Study

4. Prototype

After comparison, the team selects **Python with Flask**, because it is:

•Very easy for UG students

•Supports fast development

•Works on all OS & strong community support

Prototype Features

They make a simple working model:

•Student Login

•Upload project file

•Faculty Review Page

•Auto-generated submission ID

5. Test (Collect Feedback & Improve)

Feedback from Students:

- "Python is very easy to understand."
- "Uploading files works smoothly."
- "We can complete projects faster."

Feedback from Faculty: The interface is clean & reviewing submissions is simple.

Improvements Made:

•Added a **Bulk Download** option

•Included **error messages** for invalid uploads

•Improved **UI layout using HTML + CSS**

Case Study

Final Outcome

The college approves **Python (Flask)** as the recommended language for mini-project development.

Benefits:

- Students learn faster
- Projects become more uniform
- Faculty grading becomes easier
- Department gets a reliable online system

Next Topic:
Unit-II
Hardware and Software