COMPUTER PROGRAMMING

LECTURE NOTES

Topic 1: Introduction to Programming

Learning Outcomes:

At the end of the lesson, student should be able to:

(a) Identify programming concepts

Introduction To Programming Language

What Is the Importance of Learning Programming Languages?

Learning *programming languages* has numerous benefits:

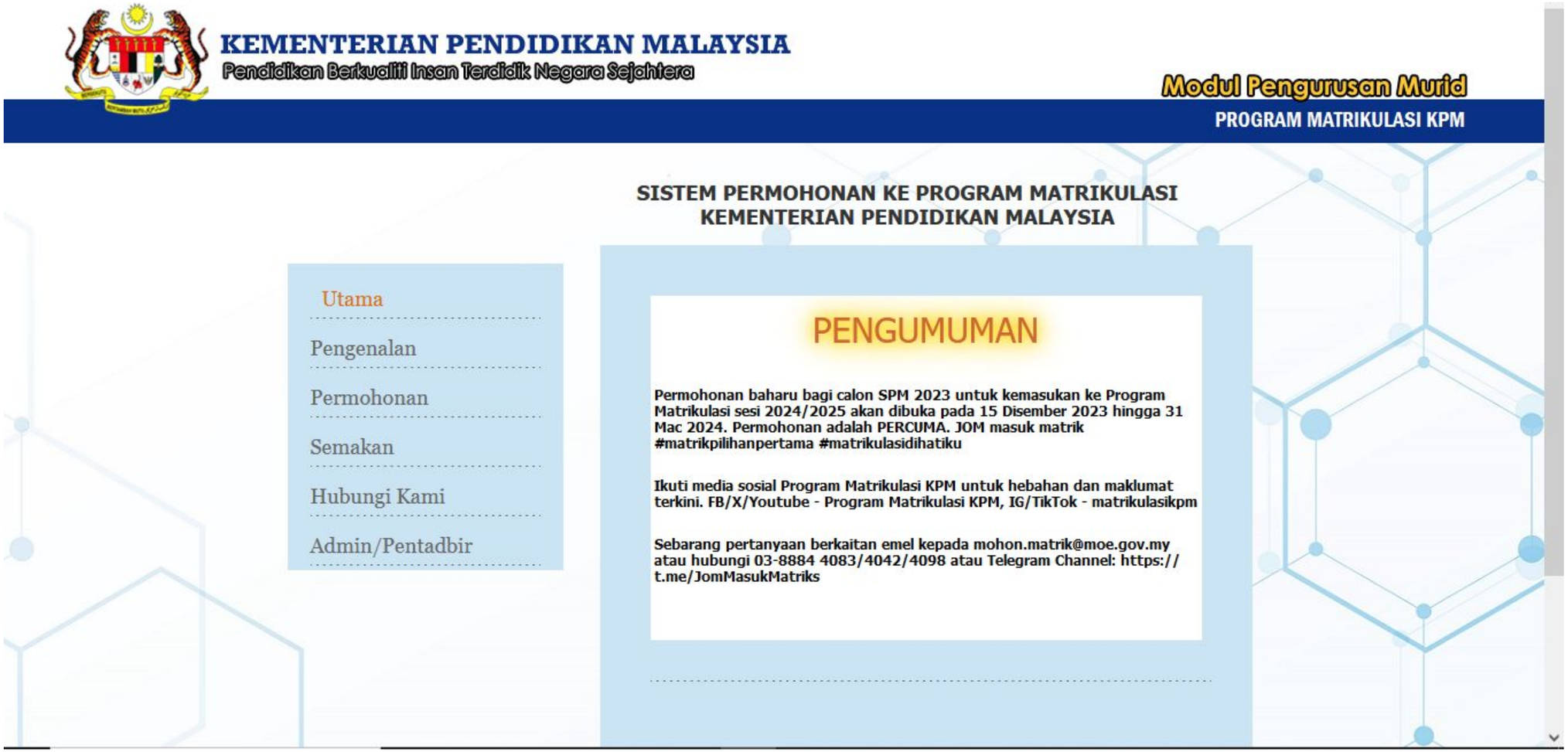
1. **Problem-Solving Skills** – Programming helps break down complex problems into smaller, manageable steps.
2. **Career Opportunities** – Coding skills open doors to various job fields, including tech, AI, and data science.
3. **Creativity & Innovation** – Programming allows you to turn ideas into real-world applications and solutions.
4. **Continuous Learning** – As technology evolves, coding keeps you learning and adapting to new advancements and improvements.

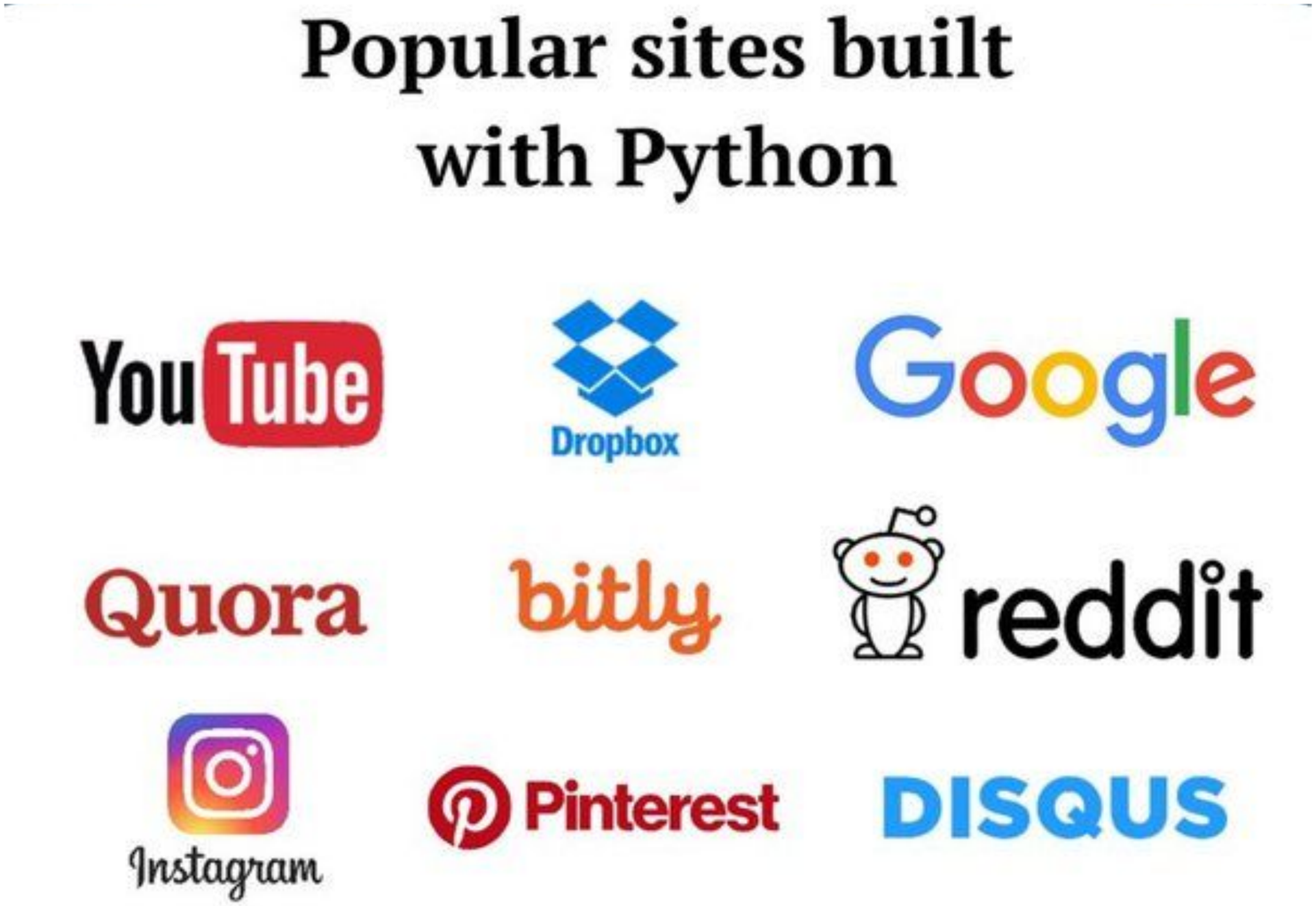
List Of Most Popular Programming Languages

Programming Language		Ratings
	Python	23.08%
	C++	10.33%
	C	9.94%
	Java	9.63%
	C#	4.39%
	JavaScript	3.71%

Source:
https://www.tiobe.com/tiobe-index/

Example Of System Developed Using Several Programming Languages




Popular sites built
with Python
YouTube
Dropbox
Google
Quora
bitly
reddit
Instagram
Pinterest
DISQUS

Why is Python So Popular?

1. Python is easy to learn
2. Python has an active, supportive community
3. Python is flexible
4. Python offers versatile web-development solutions
5. Python is well suited to data science and analytics
6. Python is efficient, fast and reliable
7. Python is widely used with IoT Technology
8. Python empowers custom automation

COMPUTER PROGRAMMING

LECTURE NOTES

CP115

Differences between programming code between Python, Java and C

Feature	Python	JAVA	C
Ease of Learning	✅ Very Easy (Simple Syntax	❌ Moderate (Verbose & Strict)	❌ Hard (Complex Syntax)
Syntax	Simple & Readable (`print()`)	Verbose & Strict (`System.out.println()`)	Complex & Manual (`printf()`)
Compilation	Interpreted (Slow Execution)	Compiled + JIT (Faster)	Compiled (Very Fast)
Speed	⏳ Slower than Java & C	⚡ Faster than Python	🚀 Very Fast (Low-Level)
Paradigm Support	Multi-paradigm (OOP & Functional)	Object-Oriented (OOP)	Procedural
Usage	AI, Web, Data Science, Automation	Enterprise, Android Apps	System Programming, OS
Code Example	print("Hello, World")	System.out.println("Hello, World");	printf("Hello, World");

COMPUTER PROGRAMMING

LECTURE NOTES

CP115

Differences between programming code between Python, Java and C

PYTHON	JAVA	C

print("Hello")
 	
Output:- Hello
 	
Note:- Python is future because it's fewer lines of code basis languages and It's have many library to solve different types of tasks.
	class HelloWorld {
	public static void main(String[] args) {
	System.out.println("Hello");
	}
	}
	
	Output:- Hello
	
	Note:- Java is very secured and rubust language that's why it's very popular than others languages.
		#include<stdio.h>
		int main()
		{
		printf("Hello");
		return 0;
		}
		
		Output:- Hello
		
		Note:- C is mostly used to develop system software.

Why is it important for a programmer to develop their problem solving skills?

- Problem-solving is the core of computer science.
- Programmers must:
  - Understand how humans solve a problem.
  - Translate input to output in a way a computer can process.
  - Write the correct syntax for the computer.
- Machines may solve problems differently than humans.

LECTURE NOTES

Topic 1: Introduction to Programming

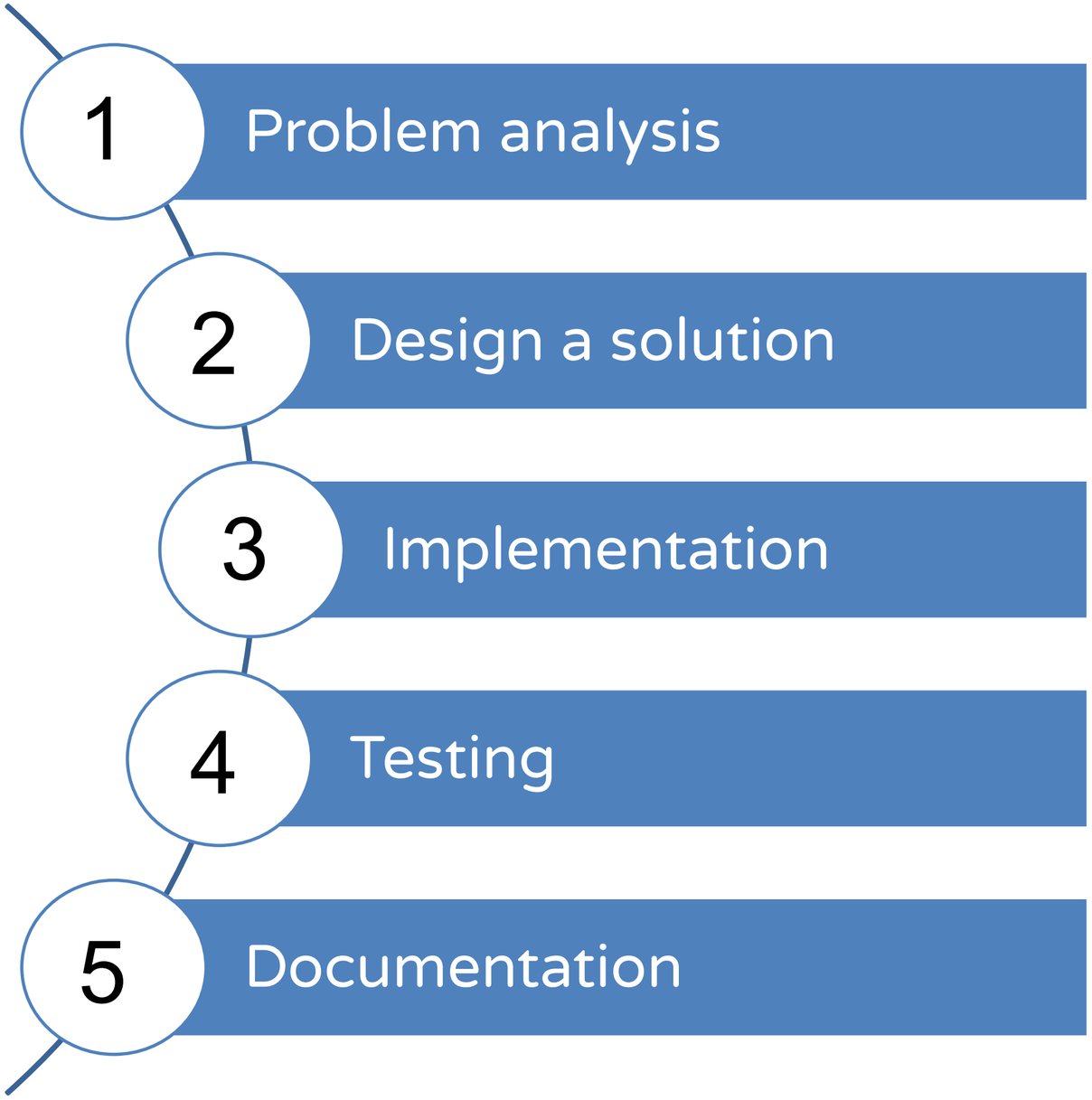
Learning Outcomes:

At the end of the lesson, student should be able to:

(b) List five (5) steps in problem solving

LECTURE NOTES

Five (5) steps in problem solving



Five (5) Steps in Problem Solving

1. Problem Analysis

- Problem analysis is the act of identifying **input**, **process** and **output (IPO)**.
- Programmers use **IPO model to** organize and summarize the results of the given problem.

Five (5) Steps in Problem Solving

1. Problem Analysis using IPO model

- IPO model
  Helps break down the problem to understand what is needed.

Input (I)	Process (P)	Output (O)

Five (5) Steps in Problem Solving

2. Design a Solution

- Design a solution involves creating an **algorithm**
- Algorithm is **logical sequence of steps** to solve a problem
- The logical sequence of steps in algorithm are derived from the input, process and output of the problem analysis step.

Five (5) Steps in Problem Solving

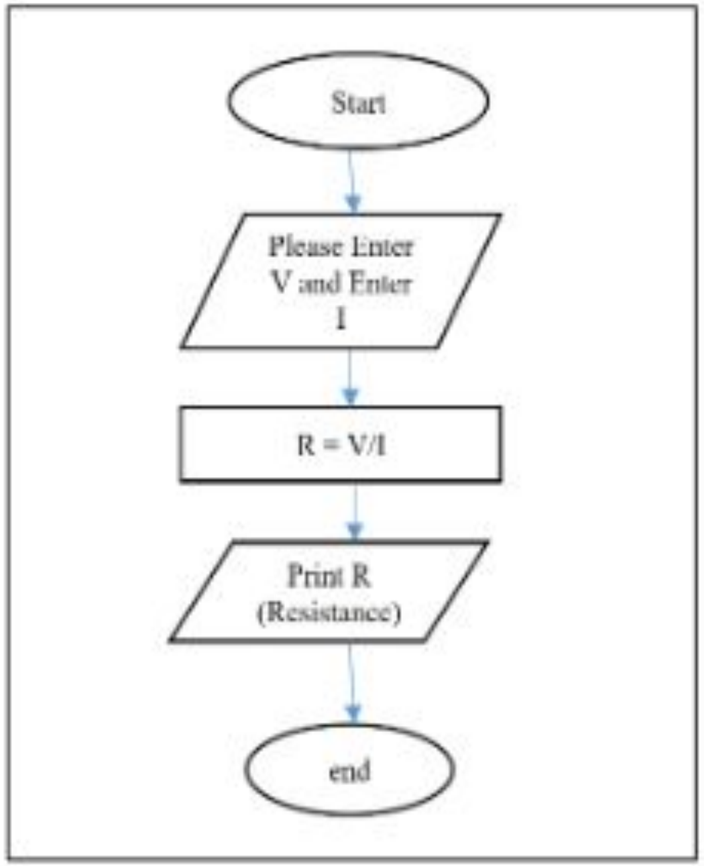
2. Design a Solution

- There are two (2) types of algorithm representations:
  - **Pseudocode** -Written steps in simple language.
  - **Flowchart** - Visual diagram of the steps.

Pseudocode

```
Start
        PRINT "Please enter the Voltage"
        GET V
        PRINT "Please enter the Current"
        GET I
        R = V/I
        PRINT "Resistance = "
Stop
```

Flowchart



Five (5) Steps in Problem Solving

3. Implementation

- Implementation is applying the algorithm by writing a **computer program** using a programming language

  - Python
  - Java
  - PHP
  - C++
  - Kotlin
  - C#

Five (5) Steps in Problem Solving

3. Implementation

- Example Python coding

```python
# Ask the user to enter three numbers
num1 = float(input("Enter the first number: "))  # Get first number
num2 = float(input("Enter the second number: "))  # Get second number
num3 = float(input("Enter the third number: "))  # Get third number

# Calculate the average
average = (num1 + num2 + num3) / 3

# Display the result
print(f"The average of {num1}, {num2}, and {num3} is: {average:.2f}")  # Display result
```

Five (5) Steps in Problem Solving

4. Testing

- Ensure the program **works correctly** and fixes errors (bugs).
- **Three types of errors:**
  - ✅ **Syntax Errors** – Incorrect coding grammars
    - Example: missing brackets ( )
  - ✅ **Logic Errors** – Incorrect calculations or wrong program behavior.
    - Example: incorrect average calculation
  - ✅ **Run-time Errors** – Errors that occur while the program is running
    - Example: division by zero

Five (5) steps in problem solving

4. TESTING (cont)

- There are three types of errors:
  - **Syntax errors -** *error in the source code of a program*

    example : missing a bracket
  - **Logic errors -** *mistake in a program's source code that results in incorrect or unexpected behavior.*

    example : incorrect average calculations
  - **Run-time errors -** *an exception error that occurs while the program is running after being successfully interpreted. Are often found during the debugging process before the software is released.*

    example : division of zero

Five (5) Steps in Problem Solving

5. Documentation

- A detailed **write-up** explaining how the program works.
- Helps future programmers **understand, maintain, or update** the code.
- Typical program **documentation materials** include:
    - the origin and nature of the problem
    - a brief narrative description of the program
    - data-record descriptions
    - program listing
    - testing results