visual programming

# Recursion

Bachelor of Information Systems
Institut Teknologi Del

# Learning Objective(s)

*This material should address the following question(s).*

- What is recursion?
- How to develop a recursive function?

# Discussion Point

**Recursion**:
The Core Concepts.

# “Images” of Itself

Sierpinsky triangle

Tree branches

Matryoshka dolls

# Question

What is **recursion**?

# Definition


***Recursion*** *is a problem-solving technique to answer a problem that consists of "images" or "copies" of itself.*

*Usually, the implementation is in the form of a function that calls itself from the inside its body.*

*Recursion calls form a stack of function calls.*

# Recursion: The Base Case

- A recursive function invokes itself to solve the problem.

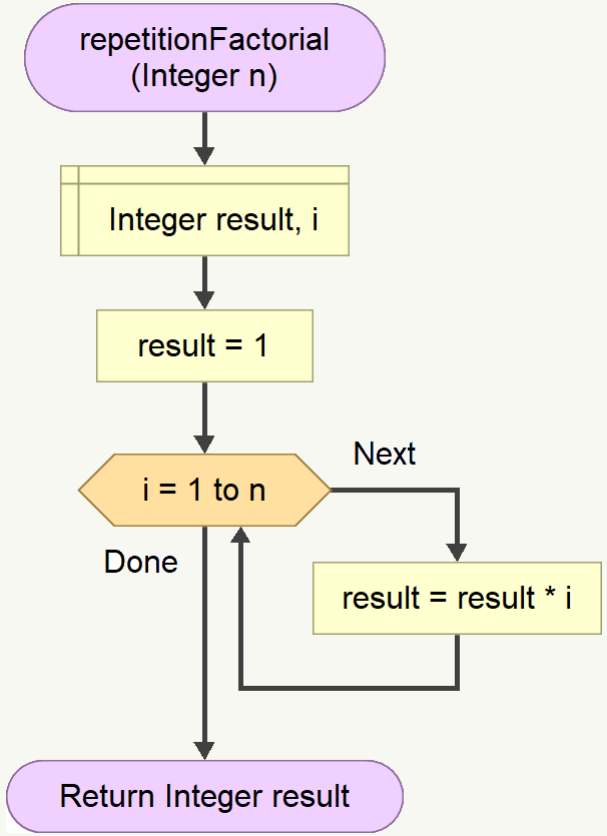

- **Question**: when should the self-invocation stop?
  - A non-stopping recursion will overflow the memory.

- The recursive function has to decide **when** to stop.
  - **Base case** is the condition which eventually stops the recursion.

# Factorial

- A solution for factorial problem can be drawn using repetitive strategy.

$$n! = 1 \times 2 \times \cdots \times n$$

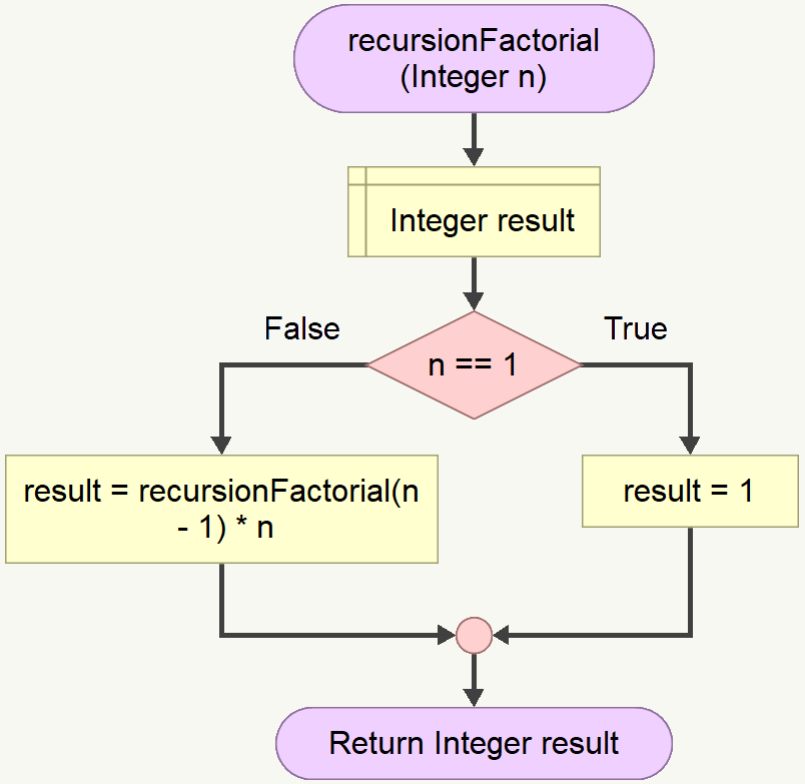


# Factorial

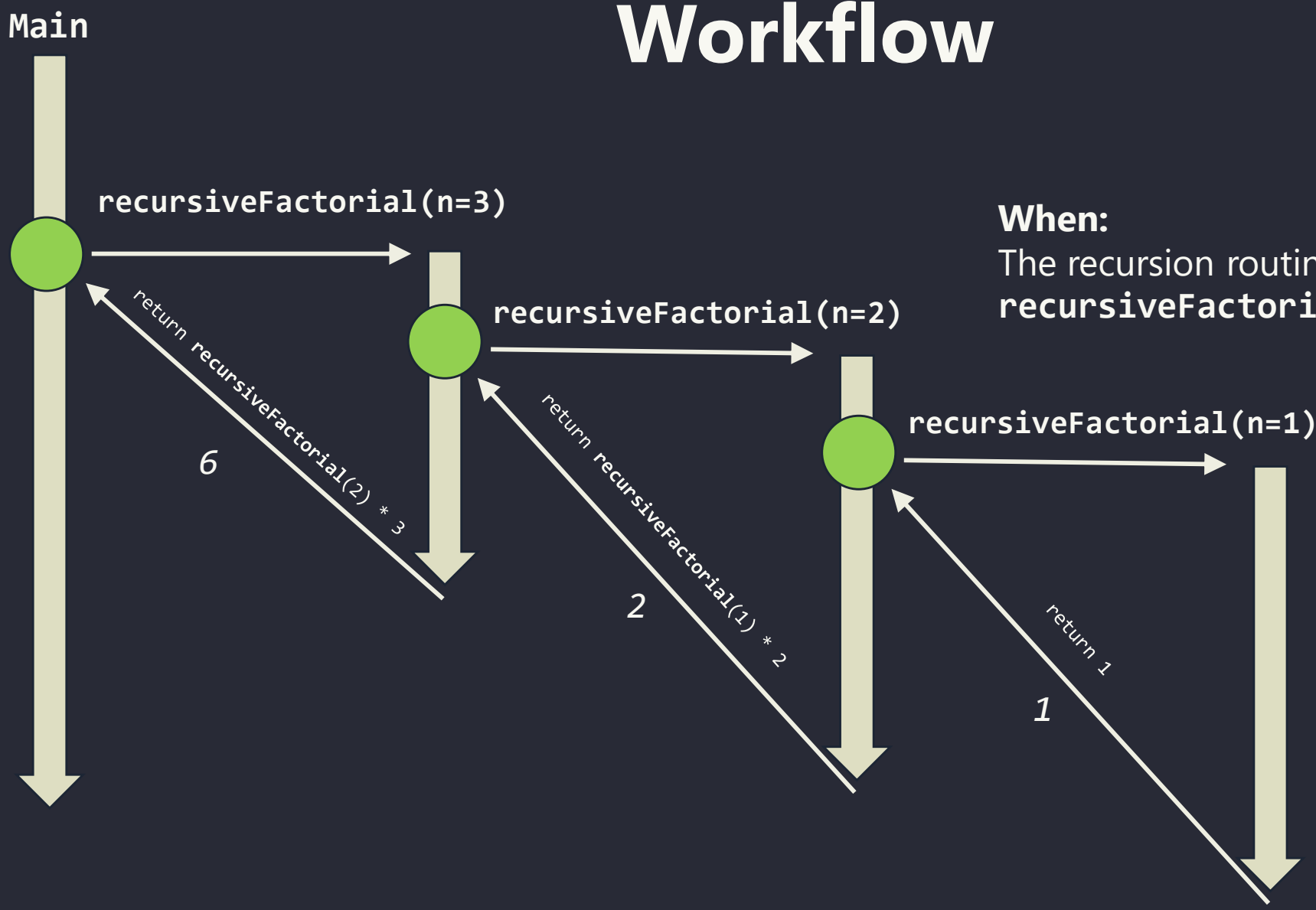
- However, the problem can also be answered using recursion approach.

$$n! = 1 \times 2 \times \cdots \times n$$

- The base case is when $n = 1$.



# Workflow




**When:**
The recursion routine finally stops when **`recursiveFactorial`** is supplied with `n = 1`.

# Repetition vs. Recursion

- Efficient (low memory).
- Using constructs, such as: `for`, `while`, `do-while`.
- Favored more in the imperative languages.

- Fast (CPU usage, parallelism).
- A recursive function invokes itself until the base case is hit.
- Uses `stack` data structure.
  - Beware of memory overflows.
- Favored more in the functional languages (Go, Lisp, etc).
- Special cases or challenges.

# Final **Thoughts.**

# Conclusion



1. A recursive function invokes itself to solve the problem.
2. A recursion must stop one time in the future.
   - A non-stop recursion will surely flood the memory (overflow).
   - A base case is the final expected stopping point.

# References

*Wassberg, J. (2020). Computer Programming for Absolute Beginners. Packt.*

– E O F –

# Course Lecturer

Mario E. S. Simaremare
Institut Teknologi Del

@simaremare
@dasar-pemrograman

# Supported by

Kementerian Pendidikan, Kebudayaan,
Riset, dan Teknologi RI

Inovasi Modul Digital 2022