PERFORMANCE-ORIENTED COMPUTING

Introduction

GOALS

- In this VU, our goals are for you to
  - Become more aware of what program “performance” ***is***
  - Understand how it can be **accurately measured**
  - Learn how to **interpret** these measurements,
    and **analyze** a program’s performance characteristics
  - And finally, how to **optimize** performance

→ *And do all of that with real-world applicability*

PREREQUISITES

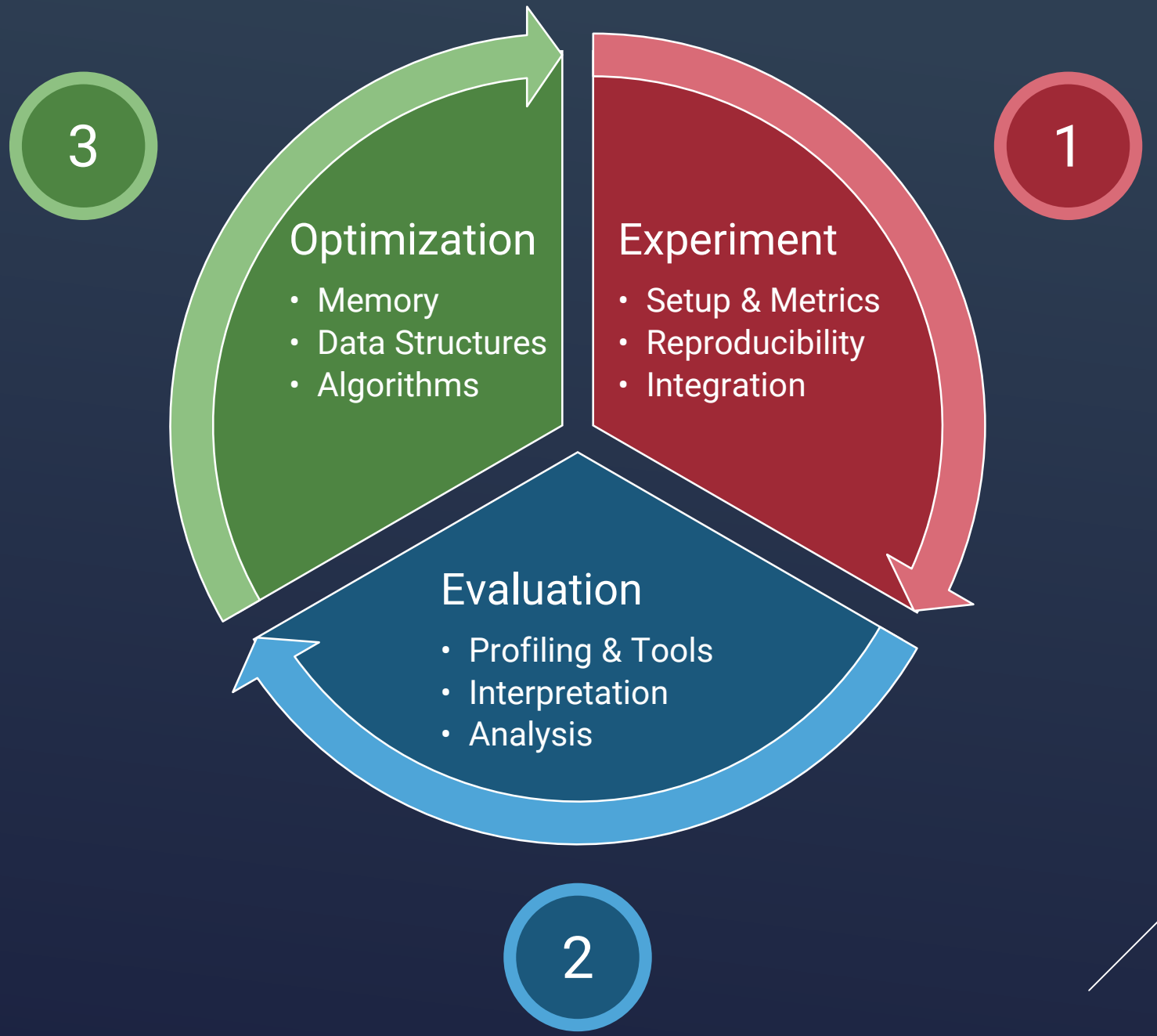
- A good understanding of the C programming language
  - You need to be able to **consistently** write **functionally correct** programs before you should worry about performance

- Some knowledge of
  - **Algorithms**, in order to understand higher-level optimization
  - **Data Structures**, so that we can have a conversation about various options and how they affect performance
  - **Computer Architecture**, in order to understand the lower-level aspects of Performance-Oriented Computing
  - **Parallelization**, at least in basic terms, as it is deeply interconnected with performance on modern HW

GRADING

- Grading will be based on
  - **40% -** Completed exercises (quality & quantity)
  - **20% -** Presentation of your results
  - **40% -** A final test
- **Additive**
- **No** additional attempts for the test

OVERVIEW


3
Optimization
• Memory
• Data Structures
• Algorithms
1
Experiment
• Setup & Metrics
• Reproducibility
• Integration
Evaluation
• Profiling & Tools
• Interpretation
• Analysis
2

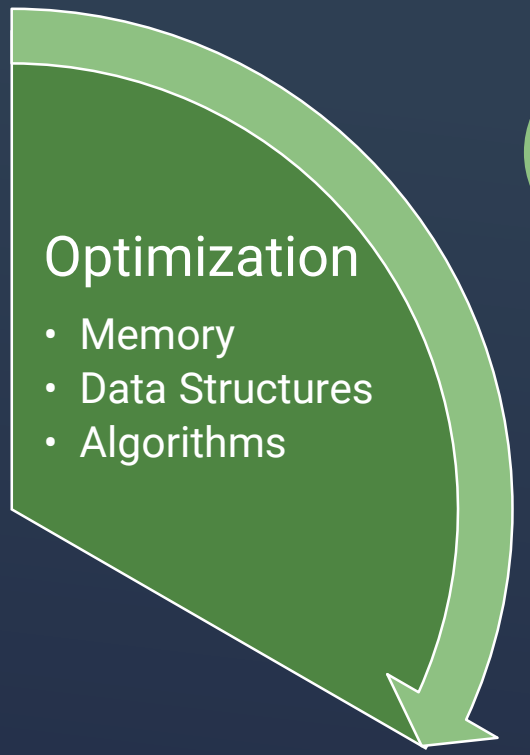
ACCURATE EXPERIMENTATION

- What to measure? Metrics, time scale, …
- Creating a reproducible experimental setup
  - External load
  - Meaningful Repetition
  - How to aggregate
- Performance integration testing

EVALUATING PERFORMANCE

- Using a Profiler
  - Types of Profiling
  - Understanding flat text profiles
  - Useful graphical representations
- Hardware Background
- Why only loops (and recursion) are interesting
- Analysis aids beyond time measurements

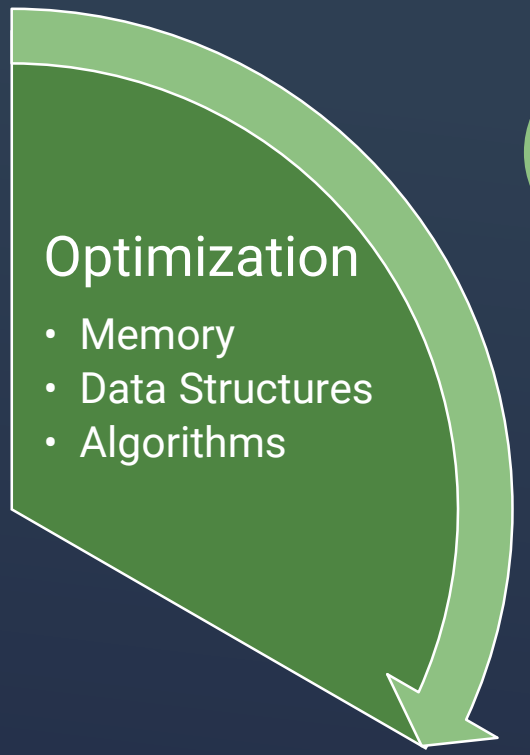
OPTIMIZATION (1) – MEMORY



3

- Computer architecture facts regarding cache vs. programming practice
- Develop an awareness for the cost of latency
  - In different use cases and parts of an application
- Dynamic memory management options and costs

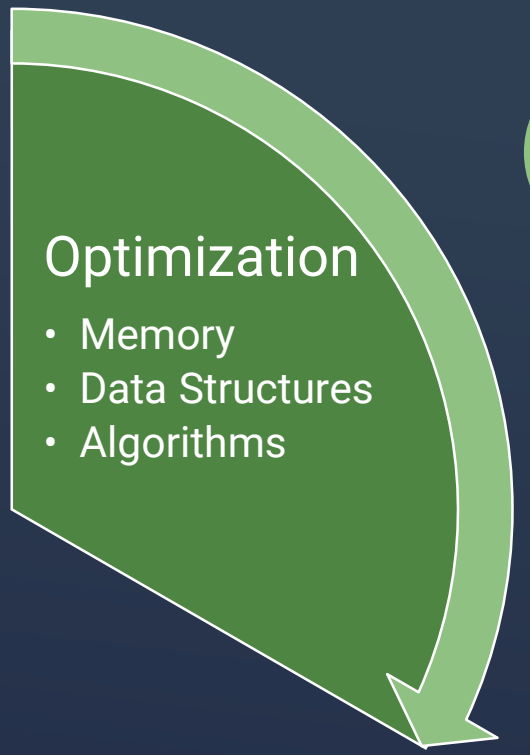
OPTIMIZATION (2) – DATA STRUCTURES



3

- Broad overview of available general-purpose data structures
- Which data structures favour which types of operations?
- What other decision criteria are there?
- Dispel myths regarding the real-world comparative efficiency of various data structures

OPTIMIZATION (3) – ALGORITHMS



3

- The relation between algorithms, data structures and performance
- Memoization
- Trade-off between execution time and memory space in various scenarios/architectures
- Sequential and parallel algorithms

QUESTIONS ?