SimpliPy: A notional machine for learning Python

Venkatesh Choppella
Moida Praneeth Jain, Gnaneswar Kulindala,
Prabhav Shetty, Anushka Srikanth

International Institute of Information Technology, Hyderabad

Outline

Background and Motivation

SimpliPy Notional Machine

Workshop Objectives and Tasks

Format

Invitation

Let's Get Started!

Topic

Background and Motivation

SimpliPy Notional Machine

Workshop Objectives and Tasks

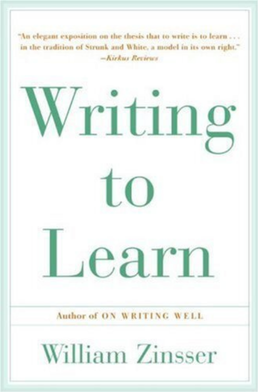
Format

Invitation

Let's Get Started!

The 3 R's of Learning

- Reading
- 'Riting
- Reasoning



"An elegant exposition on the thesis that to write is to learn . . . in the tradition of Strunk and White, a model in its own right."
—*Kirkus Reviews*

Writing to Learn

Author of ON WRITING WELL

William Zinsser

Figure: Zinsser, Writing to Learn, 1988

What does it mean to learn a programming language?

1. Comprehension: (Nelson et al. 2017)

2. Composition:

The "Central Dogma" of Computer Science

- A **program** is a sequence of instructions.

- A programming **language** specifies what is a program.

- A program runs on a **machine**.

- **Computation** is what is observed/produced when a program runs on a machine.

What is a machine?

1. Input
2. State
3. Output
4. Transitions

Levels of Abstraction

1. Hardware
2. Abstract Machine
3. Notional Machine

Topic

Background and Motivation

SimpliPy Notional Machine

Workshop Objectives and Tasks

Format

Invitation

Let's Get Started!

What is a notional machine?

1. Idealised, conceptual computer . . . implied by . . . the programming language . . . (du Boulay et al. 1981)

2. Notional Machine should be part of programming education (Sorva 2013)

Notional Machines: from Informal to Formal

1. Notional machines have been mostly visual and informal (Fincher et al. ITICSE WG 2020)

2. What about notional machines derived from the formal semantics of a programming language? (Guzdial et al. Dagstuhl 2019)

What is SimpliPy?

1. A **sublanguage** of Python designed to simplify learning

2. A notional **machine** designed to run SimpliPy programs

Topic

Background and Motivation

SimpliPy Notional Machine

Workshop Objectives and Tasks

Format

Invitation

Let's Get Started!

Research Questions

1. How comfortable are teachers using SimpliPy to teach Python?
2. How well can learners use SimpliPy to demonstrate how Python programs run?
3. What student misconceptions does SimpliPy help clear?

Tasks

1. **Identify** lexical blocks and variable declarations
2. **Construct** control transfer functions and Control Flow Graph
3. **Trace** the execution of a notional machine
4. **Draw** execution diagrams

Categories of Python programs we will explore

1. Sequential
2. Conditional
3. Iterative
4. Procedural

Future versions of SimpliPy (coming soon!)

1. Exceptions
2. Classes and Objects
3. Modules
4. Concurrency

Topic

Background and Motivation

SimpliPy Notional Machine

Workshop Objectives and Tasks

Format

Invitation

Let's Get Started!

Workshop Format

1. Tutorial
2. Worksheets
3. Feedback
4. Discussion

Format: Day 1

Introduction	4:30pm	4:45pm	15
Worksheet 0	4:45pm	5:00pm	15
Transition Systems	5:00pm	5:15pm	15
Sequential SimpliPy	5:15pm	6:00pm	45
Worksheet 1	6:00pm	6:15pm	15
Conditional SimpliPy	6:15pm	6:30pm	15
			120

Format: Day 2

Iterative SimpliPy	4:30pm	4:45pm	15
Worksheet 2	4:45pm	5:00pm	15
Procedural SimpliPy	5:00pm	5:55pm	55
Conclusion	5:55pm	6:00pm	05
Worksheet 3	6:00pm	6:30pm	30
			120

Topic

Background and Motivation

SimpliPy Notional Machine

Workshop Objectives and Tasks

Format

Invitation

Let's Get Started!

Reaching out to the community of Python teachers

1. Would you like to use SimpliPy to gain a deeper understanding of Python?
2. Would you use SimpliPy to teach Python in class?
3. Please stay in touch!
4. SimpliPy Repository URL:

Topic

Background and Motivation

SimpliPy Notional Machine

Workshop Objectives and Tasks

Format

Invitation

Let's Get Started!