What makes program synthesizers hard to learn for novice programmers?

EPIC DATA lab

Dhanya Jayagopal **Justin Lubin** **Sarah E. Chasins**

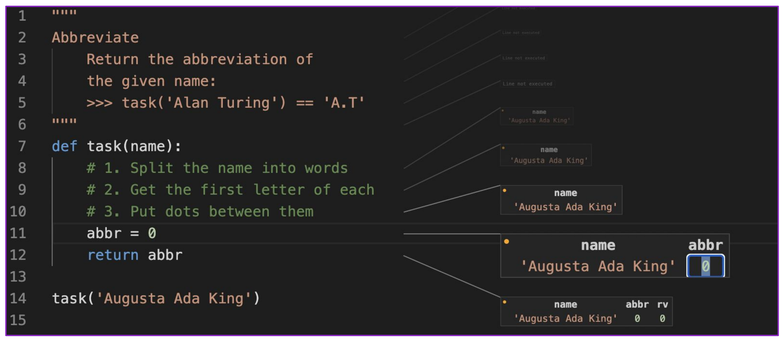
Domain experts
(Law, medicine, science, ...)

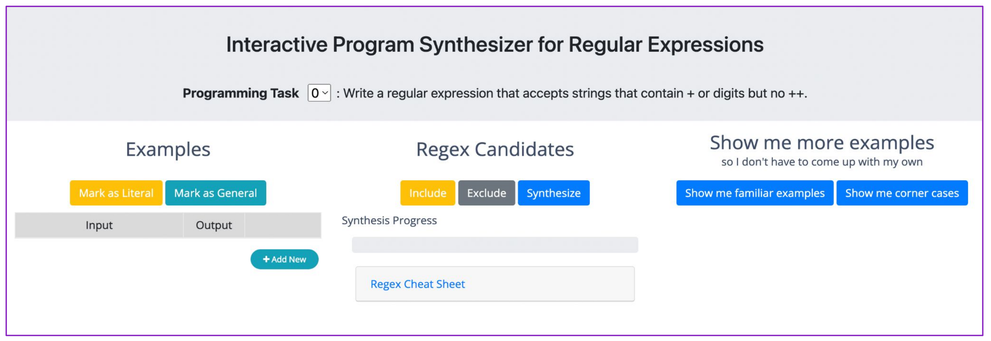
⇒

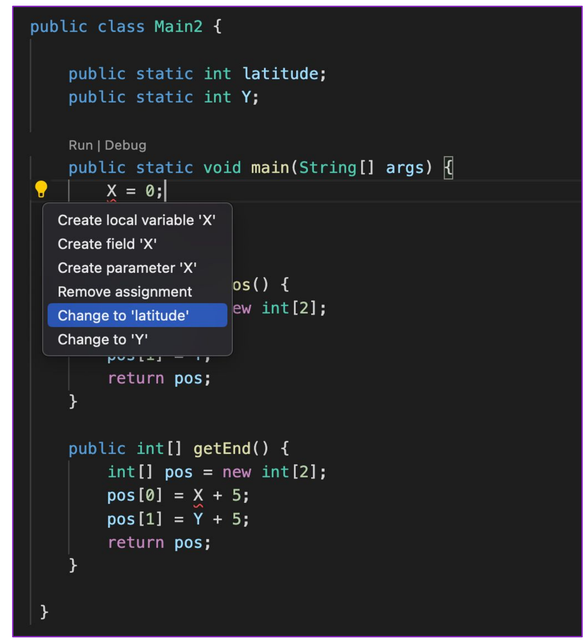
No time to learn to code!

⇒ ?

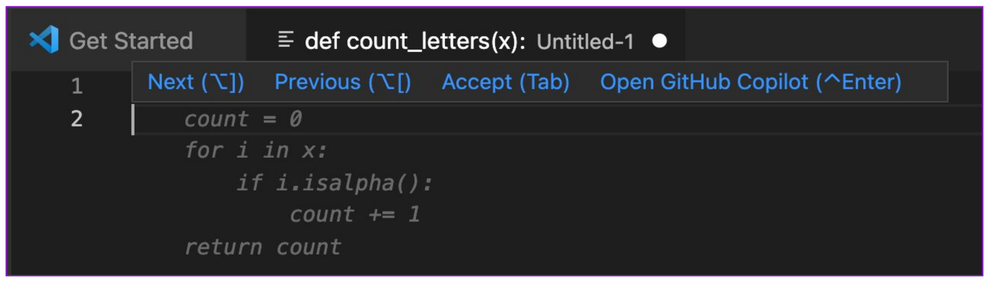
Program synthesis
Automatic generation of code that satisfies a user-provided **specification**







	A	B	C
1	**Name**	**Last Name, First Name**	**Handle**
2	Calvin Canaday	Canaday, Calvin	ccanaday
3	Perla Lindstrom		
4	Velvet Blansett		
5	Danette Giles		
6	Maxwell Herren		
7	Barry Lombardi		



Tool characteristics

Voluntary specification ⟷ Incidental specification

User-triggered initiation ⟷ Triggerless initiation

User-triggered result communication ⟷ Triggerless result communication

Understanding ***learnability implications*** of ***tool characteristics*** can help designers make ***empirically-supported design decisions***

User behavior

Incorrectly believing the synthesizer made progress

Incorrectly believing the synthesizer did *not* make progress

What makes a good specification?

Understanding ***user misconceptions*** can help designers make systems to ***proactively combat them***