What m	akes progra	IM S
	Dhanya Jayagopal	Just
Domain (Law, medicine		l lea
<pre>1 """ 2 Abbreviate 3 Return the abbreviation</pre>	Line not executed	

the given name:

def task(name):

abbr = 0

return abbr

task('Augusta Ada King')

11 11 11

>>> task('Alan Turing') == 'A.T'

1. Split the name into words

3. Put dots between them

2. Get the first letter of each

	A	В	С	Set Started	≣ def count	letters(x): Untit	led-1
1	Name	Last Name, First Name	Handle				
2	Calvin Canaday	Canaday, Calvin	ccanaday	1 Next (_])	Previous (℃[)	Accept (Tab)	Open (
3	Perla Lindstrom			2 count			
4	Velvet Blansett			for i	in x:		
5	Danette Giles			j	f i.isalpha()	-	
6	Maxwell Herren				count += 1		
7	Barry Lombardi			retur	n count		

Tool characteristics

name

name Augusta Ada King'

'Augusta Ada King'

Augusta Ada King' 0 0

Voluntary specification	
User-triggered initiation	
User-triggered result communication	

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

synthesizers hard to learn for novice programmers?

tin Lubin

Sarah E. Chasins

No time to earn to code!



Program synthesis

	active Program Synthesizer for Regula Task $0 \sim$: Write a regular expression that accepts strings that	<pre>public class Main2 { public static int latitude; public static int Y;</pre>		
Examples Mark as Literal Mark as General	Regex Candidates	Show me more examples so I don't have to come up with my own Show me familiar examples Show me corner cases	Run Debug public static void main(String[] args) { X = 0;	
Input Output	Add New Regex Cheat Sheet		Create local variable 'X' Create field 'X' Create parameter 'X' Remove assignment Change to 'latitude' Change to 'Y' position position	
Get Started	≡ def count_letters(x): Untitled-1		<pre>return pos; }</pre>	
1 Next (℃])	Previous (\[) Accept (Tab) Op	en GitHub Copilot (^Enter)	<pre>public int[] getEnd() {</pre>	
2 count =	= 0		<pre>int[] pos = new int[2]; pos[0] = X + 5;</pre>	
for i i			pos[1] = Y + 5;	
17	<pre>i.isalpha(): count += 1</pre>		<pre>return pos; }</pre>	
return			}	

- Incidental specification
- Triggerless initiation
- Triggerless result communication

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

Automatic generation of code that satisfies a user-provided specification

User behavior

Incorrectly believing the synthesizer made progress

Incorrectly believing the synthesizer did <u>not</u> make progress

What makes a good specification?



