SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

CHI 2018

Mismatch of Expectations: How Modern Learning Resources Fail **Conversational Programmers**

April Y. Wang[1], Ryan Mitts[1], Philip J. Guo[2], and Parmit K. Chilana[1]

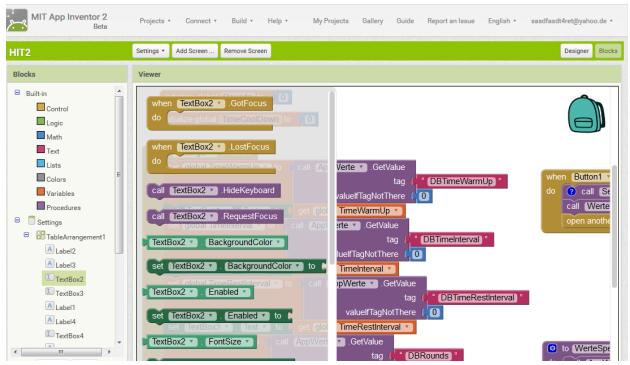
[1]Computing Science
Simon Fraser University
{ayw7, rmitts, pchilana}@sfu.ca

[2]Design Lab
UC San Diego
pg@ucsd.edu

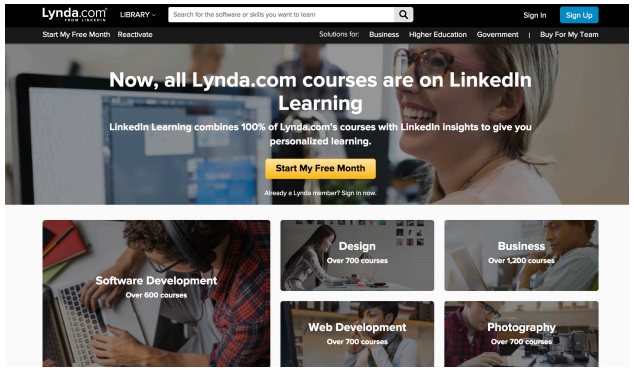
Lowering the barriers to learning programming

- Professional programmers
- CS students
- End-user programmers

Coding Workshops



Block-based Programming



MOOC courses

Increased Learner Diversity

Non-traditional populations

Designers
(Dorn et al. 2010)

High School Teachers
(Ni et al. 2012)

Older Adults
(Guo 2017)

Meet Bob: a biologist wanted to know “machine learning”

Machine learning is a hot term right now… Shall I adopt this approach in my study and put the term “machine learning” in my proposal?

Bob, Biologist

writing a grant proposal
hiring developers to analyzing data

Alice, Developer
Bob’s friend

Meet Bob: a biologist wanted to know “machine learning”

Hey Alice, I want to learn about “machine learning”, where should I start?

Bob, Biologist

writing a grant proposal
hiring developers to analyzing data

Alice, Developer
Bob’s friend

Meet Bob: a biologist wanted to know “machine learning”

Bob, Biologist

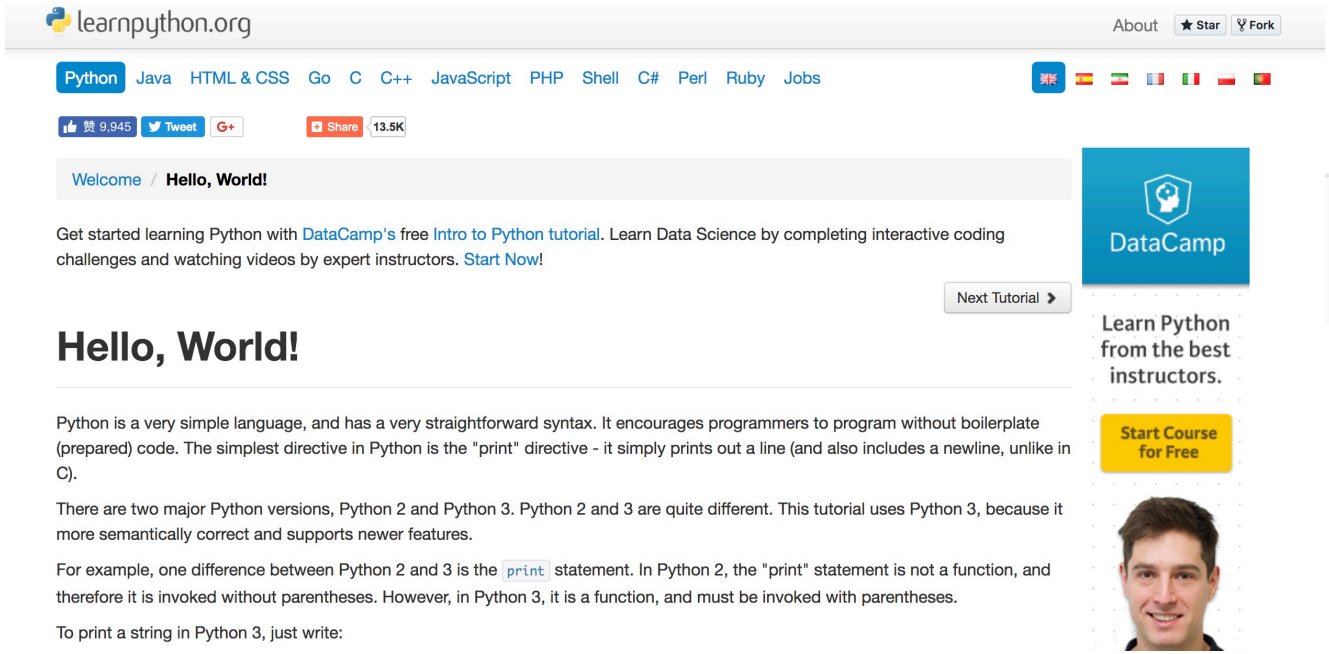
writing a grant proposal
hiring developers to analyzing data

Maybe you should learn some Python first…

Alice, Developer
Bob’s friend

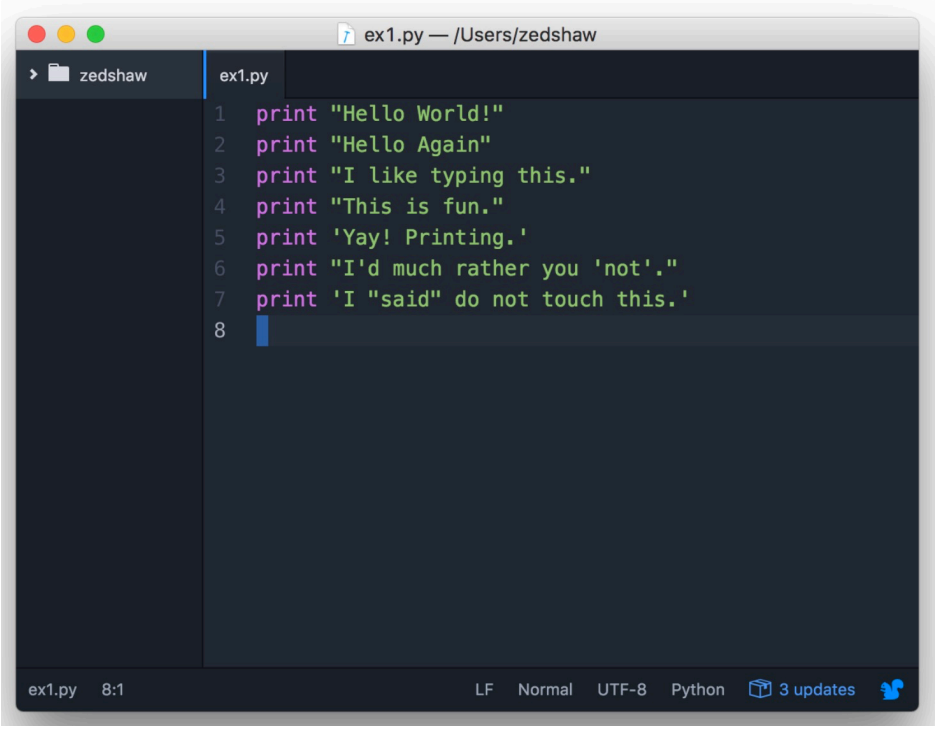
Meet Bob: a biologist wanted to know “machine learning”

Bob, Biologist



Meet Bob: a biologist wanted to know “machine learning”

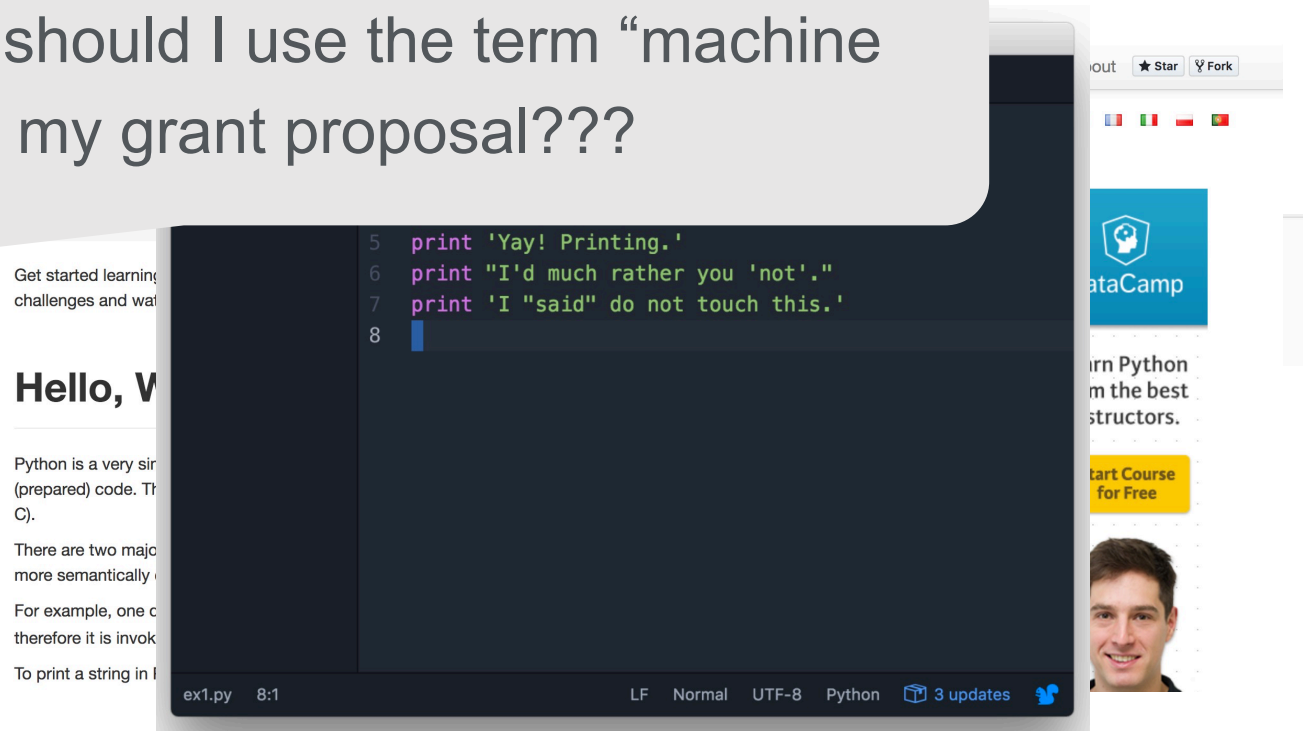
Bob, Biologist



Meet Bob: a biologist wanted to know “machine learning”

Bob, Biologist

Wait… So should I use the term “machine learning” in my grant proposal???



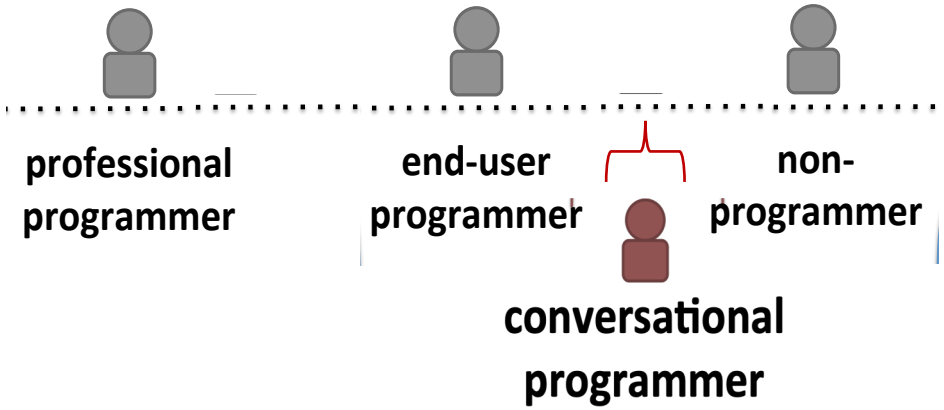
Conversational Programmer

People who are not required to write code on the job,

but motivated to learn programming

to engage more effectively in technical conversations

or to improve their marketability



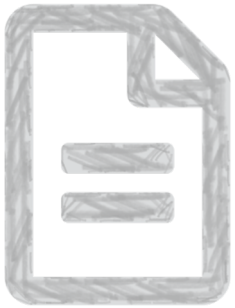
(Chilana et al. 2015, 2016)

Research Questions

How do conversational programmers actually approach programming when their goal is not to write code?

To what extent are conversational programmers' learning approaches similar or different from other non-traditional learners?

To what extent do conversational programmers find it useful to learn programming to improve their technical conversations?



Methodology

semi-structured interviews, 23 participants

Recruiting Criteria

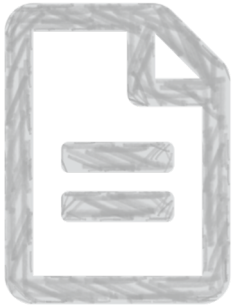
Not have a formal degree in CS

Not required to write code on the job

Have tried to learn programming

Participants represented a broad range of professions

- Entrepreneur
- Marketing coordinator
- Business assistant
- HR coordinator
- Product manager
- Advertising manager
- Artist
- Bank clerk
- Health scientist
- Psychologist
- Humanities scholar
- Library archivist
- Pharmacist
- Helpdesk support
- Medical instructor
- University staff



Semi-structured Interviews

- Challenges in technical conversations
- Approaches for learning programming
- Perceptions of the learning process

Challenges in technical conversations

Understanding the **context of conversations**

Building **rapport**

Challenges in technical conversations

1. Understanding the context of conversations

- Found it difficult to follow along technical conversations
- Did not have a shared vocabulary

An advertising manager who was struggling to interpret the data that the development team collected for campaign planning described her challenge:

*... especially when they mention terminologies around **network, database, big data, and algorithms**... I feel like I have to learn from the beginning, and that's why I am learning **Python** right now. (P9, advertising manager)*

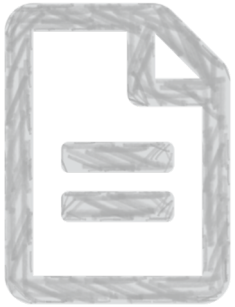
Challenges in technical conversations

2. Building rapport

- Felt ignored because of their lack of programming knowledge
- Believed they could gain more respect and credibility

A business development manager whose job was to provide customer feedback to developers said:

… if you can write code or you can understand code, ***developers respect you more****… when you're having a conversation it's easier for you to get what you want. (P7, business development manager)*



Semi-structured Interviews

- **Challenges in technical conversations**
- **Approaches for learning programming**
- **Perceptions of the learning process**

Approaches for learning programming

Did not know where to start the learning process

Ask recommendations from an expert programmer

Explore modern learning resources

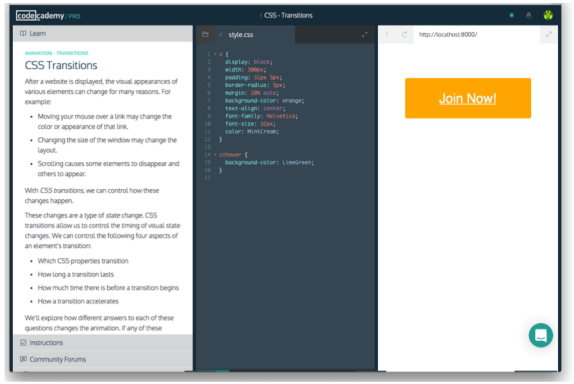
Formal approaches

- In-person courses
- Bootcamps & workshops
- MOOCs

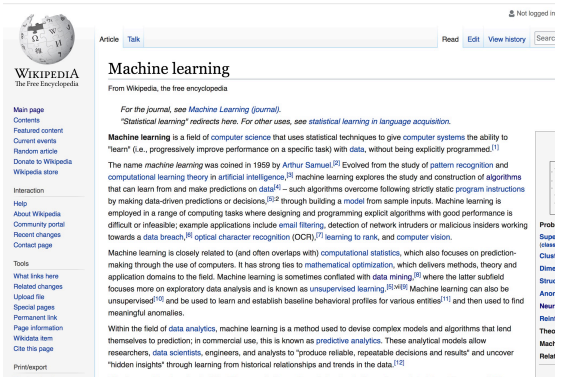
Informal approaches

- Online reference resources
- Forums
- Online coding tutorials
- Popular press

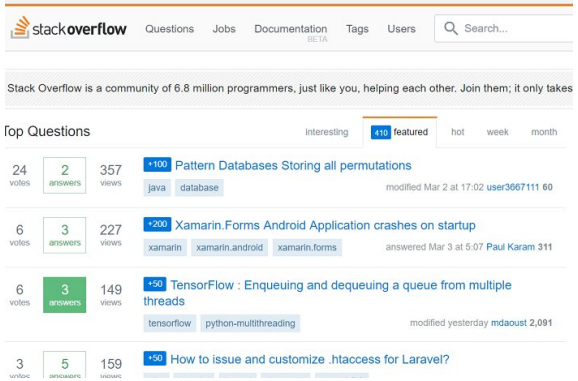
Informal approaches



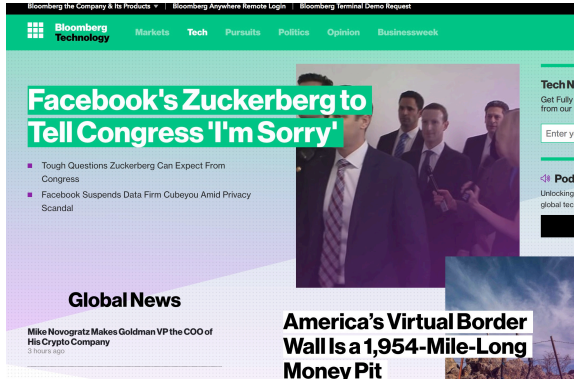
Online coding tutorials
(e.g., Codecademy)



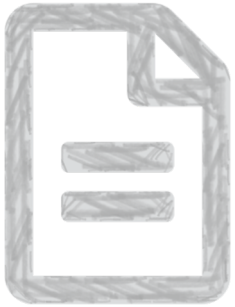
Online reference resources
(e.g., Wikipedia)



Forums
(e.g., Stack Overflow)



Popular press
(e.g., Bloomberg)



Semi-structured Interviews

- Challenges in technical conversations
- **Approaches for learning programming**
- Perceptions of the learning process

Modern learning resources FAIL conversational programmers!

Perceptions of the learning process

01 Takes too much time

02 Too much focus on syntax and logic

03 Explanations are not relevant

04 Difficult to assess the content’s reliability

05 Feelings of social isolation

06 Easy to forget details without a direct application

Perceptions of the learning process

01

02

03

04

05

06

Takes too much time

- Limited time to learn programming
- A concern raised by most of the participants

Most participants ended up being busy with their day job and found it difficult to **maintain focus and commit time for completion**:

*…This is my fourth time taking the CS50, or **fourth time attempting to**... Every time I get caught up with other work or I'm too busy. (P7, business development manager)*

Perceptions of the learning process

01

02

03

04

05

06

Too much focus on syntax and logic

- Preconceptions: to start from the beginning
- The majority (18/23) had learned a specific programming language
- Not many participants found it helpful

P11 admitted that going through the online coding tutorials did not help so much with **understanding the whole picture**:

*What I definitely needed is to be able to talk…just being able to write code, I find that I am **missing out on some kind of larger understanding**. (P11, library archivist)*

Perceptions of the learning process

01

02

03

04

05

06

Explanations are not relevant

- Seeking conceptual and application-related explanations
- Understanding the limitations and benefits of technology choices

Such explanations were **not always available** :

*… when I am learning about **cache and cookies** [on online documentation], I don't want to know if I have to use "loop" or "if-else" or anything, I want to know **what it can do for me**. (P9, advertising manager)*

Perceptions of the learning process

01

02

03

04

05

06

Difficult to assess the content's reliability

- "Trial and error" didn't work
- Needed the most credible and accurate explanations

Conversational programmers **did not often trust** the search results and still **wanted confirmation** from colleagues or friends:

*There is so much garbage on the internet that if you search something that does not look like **a credible website** then I want to verify it with a human being.(P1, entrepreneur)*

Perceptions of the learning process

01
02
03
04
05
06

Feelings of social isolation

- Domain experts in a non-technical role
- The target learners were perceived to be more experienced or even professional programmers

Participants' general perceptions about **Stack Overflow** were **negative**:

*[Stack Overflow] They're often populated by developers, **not for the lay person**… It can be pretty **toxic**. Some people are even like "Okay, this is not the place you should ask". (P13, non-tech product manager)*

Perceptions of the learning process

01
02
03
04
05
06

Easy to forget details without a direct application

- Tended to forget over time
- Only for a short-term project or to satisfy an immediate need

One participant who tried **Codecademy** to learn JavaScript said he would not do it again because he kept forgetting the concepts without applying the knowledge:

*Programmers learn and write code on a regular basis. But if you don't use it, you just **forget it**. (P17, non-tech product manager)*

Perceptions of the learning process

01 Takes too much time

02 Too much focus on syntax and logic

03 Explanations are not relevant

04 Difficult to assess the content’s reliability

05 Feelings of social isolation

06 Easy to forget details without a direct application

A mismatch of expectations when learning programming

Expectations

Improve technical conversations
Improve marketability

About **75**% of the participants expressed feelings of failure!

Do not get much benefit from investing the time and effort on these programming resources that focus on **artifact-creation**

The paradox of learning programming?

The majority (19/23) still want to keep learning programming in the future if appropriate learning resources were available

> *I will definitely **keep learning** [programming] in the future… But **I don't want to start everything from scratch**, it's like a deep pool. I only want to learn what's related with my project.(P9, artist)*

So far…

- Challenges in technical conversations
- Approaches for learning programming
- Perceptions of the learning process
  - Modern learning resources fail conversational programmers
  - The majority still want to keep learning if appropriate resources were available

How can HCI community help conversational programmers by designing tailored learning resources?

Design Implications

Can we design applications that better facilitate discovery of relevant and reliable content?

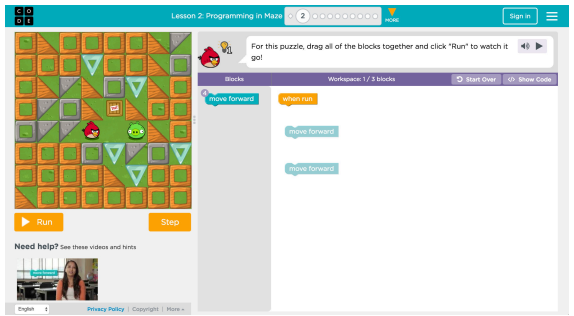
- Wikipedia-like curated overviews with small examples? How would this scale?

- Authoritativeness – Who will make contributions?

wikiHow

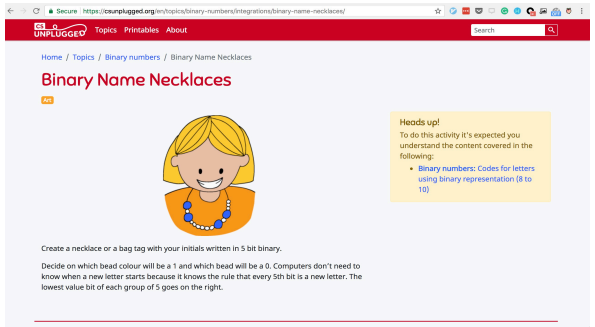
Design Implications

Can we actually teach someone useful computer science concepts *without* focusing on syntax and logic?



Block-based programming systems (e.g., code.org)

- Still largely focus on the mechanics of *programming*
- Would conversational programmers find them useful for improving technical conversations?



Conceptual instructions (e.g., CS Unplugged)

- Real-world context → programming-related concepts
- How to *talk* about a particular concept in the context of a real-world development scenario?

Design Implications

Other implications

How to design interactive high-level executive summaries that allow for more visual explorations of such concepts?

How would conversational programmers' perceptions be different if the recommendations came from other domain experts?

Summary and Contributions

- Conduct 23 interviews with a diverse set of conversational programmers
- Characterize the unique learning needs of conversational programmers
- Reveal how modern learning resources that focus on artifact-creation can fail conversational programmers
- Highlight how HCI can play a pivotal role in designing tailored learning resources

SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

NSERC CRSNG

Thanks!

April Y. Wang

ayw7@sfu.ca

Mismatch of Expectations:

How Modern Learning Resources Fail Conversational Programmers

April Y. Wang[1], Ryan Mitts[1], Philip J. Guo[2], and Parmit K. Chilana[1]

[1]Computing Science
Simon Fraser University
{ayw7, rmitts, pchilana}@sfu.ca

[2]Design Lab
UC San Diego
pg@ucsd.edu