Investigating students' preferences and perceptions of online resources in an emergency-remote introductory programming course

by

David Wong-Aitken

B.Sc., National University of Trujillo, 2008

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2021

Declaration of Committee

Name:	**David Wong-Aitken**
Degree:	**Master of Science**
Thesis title:	**Investigating students' preferences and perceptions of online resources in an emergency-remote introductory programming course**
Committee:	**Chair:** Victor Cheung Limited Term Lecturer, Computing Science
	Parmit Chilana Co-Supervisor Associate Professor, Computing Science
	Diana Cukierman Co-Supervisor University Lecturer, Computing Science
	Tenzin Doleck Committee Member Assistant Professor, Education
	Angelica Lim Examiner Assistant Professor of Professional Practice, Computing Science

Ethics Statement

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

Although online programming courses offer flexible learning formats, research shows that students report increased feelings of isolation and often end up dropping out of such courses. In this thesis, we investigated a study case of university students enrolled in an introductory programming course offered in an emergency-remote context. Through surveys and interviews, we sought to understand what difficulties students faced, what learning strategies they used, and what types of resources they preferred. We found that most students felt compelled to consult web-based resources (e.g., Q&A forums, videos) on their own and chose to ignore their own notes and curated resources offered by the instructor. Although many of the students realized that they wasted time searching the web unsuccessfully, they continued to search incessantly, neither asking for help nor self-monitoring the value of their learning strategy. We discuss several possible theories and provide recommendations for improving students' interactions with learning resources.

Keywords: novice programmers; learning strategies; introductory programming; emergency-remote learning; learning resources; self-monitoring

Dedication

For teachers all over the world.
The future owes you more than ever.

For those late bloomers whose faith in the pursuit of their dreams was never lost.

Acknowledgements

Above all, I would like to express my deep gratitude and appreciation to my supervisors, Dr. Parmit Chilana and Dr. Diana Cukierman, for believing in me and helping me fulfil this dream with enthusiastic encouragement from the very beginning. I am very lucky to have you both. I am eternally grateful.

Also, special thanks to Dr. Sheelagh Carpendale. I learnt a lot from her suggestions, talks and advice during these months in our SFU Lab meetings.

I want to thank my fellow lab members at SFU Interactive Experiences Lab: Anjali, Amir, Forozan, Laton, Narges, Rimika, Kavana, Maryam and the others who joined us later. Thank you for providing your advice and support.

My appreciation for all the course instructors who provided access and supported this thesis. Also, a special acknowledgement to all of our study participants. This thesis would not have been possible without your participation.

Thanks to Camucha. I stole some time of ours to do this thesis.

Finally, I wish to thank my parents: Higinio and Dora. Dad, thank you for being there and encouraging me to pursue my dreams. Mom, there are no words to tell you how much I love you. You will always be in my heart.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since programming is becoming a fundamental literacy skill worldwide, a plethora of formal and informal ways to learn programming has emerged over the past few years. In addition to traditional classroom instruction, many initiatives and institutions now offer new ways to learn programming online through dedicated workshops and tutorials [112] and massive online courses or MOOCs [71]. Learning programming online offers lots of flexibility and benefits as students can see course content in different interactive and multimedia formats [21]. However, studies have shown that online learners report increased feelings of isolation and face difficulty seeking help [90] [104], which often results in higher dropout rates in online courses.

With the global Covid-19 pandemic, most teaching has moved online and issues with online learning have come to the forefront. This online learning modality has been referred to as *emergency-remote teaching* [24]. This modality is somewhat different from regular online teaching. Unlike regular online courses, emergency-remote courses were created to provide temporary access to instruction and instructional support in a manner that is quick to set up and reliably available during an emergency or crisis [48]. This implied, among other things, that there was insufficient time for proper course planning, and lack of choices for students because of the emergency nature of the modality. In this new paradigm, instructors have had to adapt their in-person courses ad-hoc to a remote setting when many of these instructors had neither prior experience with online teaching nor enough time for planning. Unlike MOOCs or other online courses which some people may explore informally, college and university students did not have any choice but to stay enrolled in their institution's emergency-remote courses for full credit [53]. Initial studies have indicated that students have been experiencing increased issues of anxiety and lack of self-discipline in such courses [91].

Given the circumstances of emergency-remote teaching and the challenges associated with learning programming for the first time [13] [101], we wondered what it was like for novices to learn programming in an emergency-remote context, what types of difficulties they faced, and how they coped with these difficulties. We carried out a study to understand

the perspectives of students enrolled in an emergency-remote introductory programming course at a large North American university. We built our key research question on previous research which showed that when students learn online, they prefer using a variety of different resources and formats [99], such as videos [59] [73], Q&A forum resources [27] [63], among others.Our key research question was: *How do students make use of different online resources when learning programming for the first time in an emergency-remote course?*

In this thesis, we present insights from a case study of two consecutive emergency-remote classes in an introductory programming course, open to all students pursuing degrees in different disciplines. In the first term, we surveyed 84 students to understand how novice learners coped with emergency-remote courses while learning how to code. Based on a subset of these survey respondents, we conducted 13 follow-up interviews in order to obtain more insights about how students use and get help from learning resources, and learn more about what motivated students to seek out more resources. To further corroborate our findings, we refined our survey questionnaire and conducted a second survey with 92 students in the following term's offering of the intro programming course. Our specific goal was to understand which external resources were considered helpful and why. We were also interested in knowing what strategies students followed to obtain these resources and how they self-monitored their learning while using these resources.

1.1 Key findings

Among our key findings, we observed that:

1. We found that students' learning strategies were greatly intertwined with external web-based resources that they looked for on their own. In fact, survey1 indicated that 98% of students used at least one external resource. This was somewhat surprising as most of our survey1 respondents and interviewees found resources provided by the instructor useful and appropriate (11 out of 13 from the interview).

2. Students looked for new resources relying on a quick trial-and-error approach, trying to sift through as many resources as possible, expecting to locate specific answers with little effort. But, since these students were novice learners of programming, they had difficulties identifying what to look for and figure out what is relevant. Survey2 respondents indicated that they always used Q&A forums resources over online videos (8.9% vs 4.3%), and found Q&A forum resources extremely useful compared to online videos (14.0% vs 2.0%). This is mainly because they could match the answer from a Q&A post with their questions in a simple word-per-word approach (37.5% students in survey2 indicated that). Students used these Q&A forum resources even though they realized that these resources are more oriented to professional programmers and not to novice learners.

3. Even though many students indicated that using this approach meant wasting large amounts of time searching the web, they could not help to keep looking, being in a constant *scanning mode*. When a difficulty understanding the newly found resource arose, they blamed the resource for being too complex and jumped to the next resource, not realizing that this approach made them unable to learn properly. This behavior was fueled by students' incessant belief that it is easy to get resources from the internet on any topic. Because of this compulsive habit, many students kept moving from resource to resource, rarely stopping to reflect on what they were actually learning or applying strategies to minimize the time they wasted.

4. From the students' perspective, "instant answers" that they could find on their own were more efficient and made them feel that their learning process was under control. On the other hand, asking for help was considered to be the opposite: slow, inaccessible, and potentially high in terms of social cost. Not asking anyone when solving programming exercises was the most valued attribute (28.9% of total, survey2). 8 out of 13 in the interview indicated that asking for help would be the last resort, only reserved for extreme cases.

To better understand these findings, we draw upon some theories from Human-Computer Interaction (abbreviated HCI), information-seeking, and learning sciences that may help to explain why novice learners of programming relied extensively on external resources the way they did. We also discuss the larger implications of our findings from the perspective of HCI and CS Education for improving pedagogy and designing new tools and approaches to improve the experience of students learning programming online.

1.2 Organization of this thesis

This thesis is divided into seven chapters. The first chapter is the introduction which present an overview of our work, our motivations, our research question, and a summary of the findings of our study.

The second chapter presents other works related to our research. In this chapter, we report on prior studies about teaching introductory programming courses online, different resource formats for learning programming, challenges with self-monitoring in learning programming, how programmers use external web resources, and works related to novice learners' reluctance to seek help.

The third chapter is focused on explaining our study methods. We detail here our research site, the study procedure and instruments, and the analysis and presentation of results. We detailed students' differences between group 1 (first term in emergency-remote learning) and group 2 (second term in emergency-remote learning).

The fourth chapter is oriented to presenting our key results. We divided that chapter into five sections. These sections describe students' reliance on resources they look at on their own, the process they use to obtain new resources and assess their quality, the students perception about time and required effort to obtain new resources, and their reluctance to ask for help from their peers, teaching assistants, or instructors.

The fifth chapter is dedicated to discussing and reflecting on our key insights by drawing upon ideas and theories from prior literature. We selected key theories that may help us to understand how and why students rely extensively on external resources.

The sixth chapter provides some ideas for HCI tools and CS education approaches. We discuss three different ideas for HCI tools and two CS education approaches that may help students to focus more on solving the programming assignment and less in wasting time looking for external resources. We also provide some insights for future research that may be interesting to explore.

The seventh chapter is the conclusion. This chapter summarizes our key results, procedures, discussions, and proposals, wrapping up our work.

Chapter 2

Related work

In this chapter, we review prior studies from HCI and Computing Education that are most relevant to our research. In particular, we draw upon insights about teaching introductory programming courses online, use of different resource formats for learning programming, challenges with self-monitoring in learning programming, how programmers use external web resources, and works related to novice learners' reluctance to seek help.

2.1 Studies on teaching introductory programming courses online

As the demand for programming skills has been increasing, many innovations have been explored to offer introductory programming courses online. Unfortunately, studies have shown that such online courses have a high rate of dropouts and low engagement [90] [104] , especially in the case of MOOCs [23]. Issues of isolation, disconnectedness, and technological problems are common factors that lead students to abandon online courses [104] and particularly programming courses [30]. Many of the issues that students experience online are similar to in-person classes, such as learning problem-solving skills and converting ideas into programming expressions [19] [89].

Recent works show that individual levels of performance, emotion, skills and participation affect engagement in online courses[9]. Hence, many students have different perceptions about online courses which affects their level of commitment and participation [31] [35] [54] . However, with the ongoing pandemic, many university and college students around the world were forced into emergency-remote courses. They could not choose to switch to in-person instruction if the online medium did not suit them well.

We are only starting to learn about the characteristics and impact of emergency-remote teaching. Although remote teaching offered some level of flexibility and the ability to reach a large number of students [39], prior work has shown that creating engaging online courses can be time-consuming and costly [10]. Most instructors did not have the time or prior training to set up emergency-remote courses and faced an uphill battle [2] [48]. Students have

also been experiencing increased levels of anxiety [91] and low engagement [53] in emergency-remote courses, and this problem is emerging in intro CS courses as well [24]. Our case study adds new insights to illustrate some of the challenges that novice programmers faced in an emergency-remote context, particularly, how they made of use of different resources in their learning journey.

2.2 Studies on different resource formats for learning programming

The web offers millions of resources in different formats for learning about theoretical aspects of programming and locating real-world coding examples. For example, instructional videos are becoming a popular way for students to visually understand programming-related concepts [59][73] and are used as the main method for content delivery in nearly all massive open online courses (MOOCs) [11]. More recently, various video-based live-streaming platforms related to coding concepts are also emerging [14] [44]. However, despite the popularity of videos, recent studies have shown that 50% of MOOC certificate-holders students watched less than half of the MOOC course videos[18]. It was also found that students often disregard large segments of educational videos[41], or videos contribute little to student performance [64]. Videos are also expensive and time-consuming to produce [43].

Another resource that is becoming popular is the use of programming question and answer (Q&A) sites, such as *Stack Overflow* which potentially could allow learners to leverage the knowledge and expertise of more expert programmers in seeking answers. However, these Q&A sites were designed for software developers and their on-the-job needs, not necessarily for novice learners of programming [8] [25]. Although forums offer the ease of looking up specific technical answers, it can be difficult for end users to formulate their problems as queries [77] and another issue is that users rely too much on the exact answer and do not analyze or reflect on the content [1] [98]. Some studies indicate that using StackOverflow for undergraduates was at least as effective as traditional resources [28]. For example, reformulating a course with *"learning by doing"* and using StackOverflow as main source of reference, was reported to improve students' understanding, problem solving and positive perceptions [63]. However, these previous works mostly consider traditional face-to-face courses where the communication and help flows more easily compared to online or emergency-remote courses.

2.3 Studies on challenges with self-regulation in learning programming

Self-regulation of learning is the process in which students actively monitor and control their learning using a variety of cognitive and behavioral strategies [111]. Research in self-

regulation and self-direction when learning programming have offered insights into how students turn problems into code [38] [47], build self-motivation [103], resilience [80], and overall self-regulation [32] [42] [61]. For example, prior work has shown that novice programmers' self-regulation tends to be inconsistent and shallow [62], but that it is trainable through direct instruction[34]. Although we know that use of online resources are popular among students learning programming, we have few insights into how the use of such resources affects self-regulation. For example, some studies have indicated that online resources may in fact be able to help build self-regulation [107], and improve learning outcomes [49]. However, for novice programmers, some structure when using web resources was found helpful compared to not using any structure [51]. Successful learning with technologies is based on the premise that learners adaptively regulate their cognitive and metacognitive behaviors during learning [3]. When they fail to do so, we call it *dysregulated learning* [5]. This term is used to describe a group of behaviors that learners use that lead to minimal learning, such as failing to deploy effective learning strategies. For example, students dysregulate their learning when they fail to select the right resources for learning, according to their requirements, needs or study level. Our study adds new insights into how novice programmers self-regulate their progress when their learning is closely intertwined with a variety of web resources and the types of challenges they face. This study also shed some light on how students' learning may be dysregulated when relying excessively on web resources.

2.4 Studies on how programmers use external web resources

Although we are not aware of studies that shed light on how students make use of external web resources when learning programming online, we review complementary research on how programmers use external web resources. One of the well-established ideas comes from Pirolli and Card's Information Foraging Theory [75] [76]. For example, the theory of information foraging has been used to explain programmers' diet when using resources [50] [74], this theory was also used to explain how end-user programmers forage in online repositories [56] and to create tools to support information foraging for novice programmers [88]. Brandt's Opportunistic Programming [16] [17] aims to explain how programmers use opportunistic approach to use external resources to produce cheap and quick code to solve programming problems. Opportunistic programming approach has been used to understand difficulties in information seeking among end-user programmers [29]. One of the most important aspects of this approach is that it may help us understand why some programmers copy/paste code from internet; they may be behaving as opportunistic programmers whose code will be used few times. Our findings about students' behavior are complementary to the findings about opportunistic programmers.

2.5 Studies on novice learners' reluctance to seek help

Researchers have identified some key factors that influence students' decision to seek or avoid help in the classroom. For example, Price and Liu [79] observed that students may avoid help for practical reasons: help may be inaccessible or inconvenient, and the use of help may be prohibited or go against social norms of the classroom [85]. Students may avoid help because of concerns about the help-giver's competence (e.g. if it is a peer), a desire for independence [96], or because of a perceived threat to competence [20]. Lower-performing students and students with lower self-esteem are more likely to feel a threat to competence by seeking help, and this threat can manifest in help avoidance or seeking expedient help (with the goal of finishing the problem), rather than instrumental help to foster learning [52]. Students with a *performance* achievement-goal orientation, focused on others' perception of their relative competence, are less likely to seek help than those with a *mastery* orientation, focused on learning and self-improvement. Help-seeking in the classroom is also a social experience, and students with a social status-goal orientation, focused on social visibility and prestige, may feel increased social costs of help-seeking and therefore avoid help [84] [85]. There is also some prior work related to programmers' reluctance to seek help. For example, Begel and Simon's work [12] on boarding process to new college graduates at Microsoft observed that new software developers had the following misconception:"*I must do everything myself so that I look good to my manager*", therefore they resort heavily to external resources in order to not reveal their perceived deficiencies. Our findings shed some light about the reported choices made by novice programming learners in emergency-remote environments when deciding whether or not to seek help from others.

Chapter 3

Study methods

In this chapter, we describe our study methods. We carried out our research in two different terms in an introductory emergency-remote course, explaining the differences between our two groups of students and why we studied two group instead of just one. Our study procedure consisted of two surveys and one set of interviews conducted in the first group of students, and a survey conducted in the second group of students. Our data analysis consisted of descriptive statistics for the closed-ended questions and bottom-up inductive analysis approach for open-ended questions and the interview transcripts. Next, we describe in detail our study methods.

To understand how students in emergency-remote introductory programming courses were making use of external resources, we used a combination of descriptive methods. We conducted surveys in two different teaching terms and carried out interviews with a subset of the survey respondents from the first group. All of the study procedures and instruments received clearance from our university ethics board.

3.1 Research site

We recruited participants exclusively from an emergency-remote introductory programming course offered in the CS department of a large university in North America. This course had never been offered in a fully online format prior to the pandemic in 2020. This course was open to undergraduates from any discipline and can be considered equivalent to CS1. The core programming language taught was Python and the course lasted for 13 weeks.

Our study lasted two consecutive terms: Summer 2020 and Fall 2020. The course and the emergency-remote approach for both terms were unchanged. The course was delivered using a combination of the Canvas Learning Management System, Blackboard Collaborate, and Zoom video conferencing tool three times a week on a regular basis. The instructors published their slides before every class and posted the recordings after class. The book *How to Think Like a Computer Scientist: Interactive Edition Python*, from Runestone Interactive [67] was used.

In both the summer and fall offerings, a significant group project was assigned at the end of the semester. Almost every week, students were provided with assignments and quizzes; some were intended as practice, some for course points.

Next, we describe the particularities of each group of students.

3.1.1 Students group 1: summer 2020 term

Our first research group was made up of students enrolled in this course for the summer 2020 term. This was the first term when this course was taught using the emergency-remote format. There were 200 students enrolled in this group. They were all taught by the same course instructor, who provided programming quizzes based on materials seen in class and the official interactive textbook. Points were awarded for reading and doing practice exercises within the textbook.

Due to the summer term nature, there were some differences in the student population in comparison with a regular term. For example, enrolment is regularly lower, and typically students in this term are non-cs majors.

3.1.2 Students group 2: fall 2020 term

Our second research group was made up of students enrolled in this course for the fall 2020 term. This was the second term when this course was taught using a emergency-remote format. There were 669 students enrolled in four sections, each taught by a different instructor. The four sections were closely coordinated, to the weekly learning objectives level, and had the same assignments and tests. All four sections had a single course website and used the same Runestone textbook [67] used in the summer term. Points were awarded for reading and doing practice exercises within the textbook. Questions and exercises were created by the instructors, based on materials seen in class and in the official interactive textbook. Practice review questions embedded in the book were given points. These questions could be answered until they are correct, without penalty.

3.2 Study procedure and instruments

In this section we describe our study procedure and instruments. We used two surveys (we will call them Survey1 and Survey2, respectively) and one set of interviews. Survey1 and all the interviews were conducted in research group 1 (summer 2020 term), while survey2 was conducted in research group 2 (fall 2020 term). Before describing our study instruments, we first clarify our definitions of internal vs. external resources that we use in this thesis.

Internal vs. External Resources: We classified the resources that students described using in two groups. Any resource that was provided or suggested by the course instructor or TA was classified as an *internal resource*. Any resource that the students looked up on

Table 3.1: Type of resources and their classification. We defined an internal resource as any resource that is provided or suggested by the course instructor, and an external resource as any resource that students locate on their own.

Category	**Resource description**
Internal	Check notes or course slides
	Seek help from teaching assistant
	Use the provided interactive textbook (Runestone)
	Post question into classmates forum (Piazza or Canvas)
	Consult with the course instructor
External	Use other printed books or e-books
	Watch YouTube videos
	Read relevant web pages in general
	Use external forums (eg. StackOverflow)

their own was classified as an *external resource*. See table 3.1 for examples of both types of resources.

SURVEY1: The survey1 questionnaire consisted of 27 questions with 24 closed-ended questions and 3 open-ended. The questions were divided into four groups: basic demographics and prior programming experience, resource preference for learning, learning difficulties and strategies to cope with them, and self-monitoring strategies when learning through emergency-remote online courses. The goal was to find out how novice learners cope with new emergency-remote courses and the strategies they deploy to overcome difficulties.

We conducted survey1 in research group 1 (summer 2020 term), from weeks 8 to 12. The survey responses were anonymous, but the instructor offered minimal points for participating (0.5%). We also offered a $50 amazon gift card raffle for participation. We obtained 84 completed surveys out of 200 students (42.0% response rate of the summer 2020 class). Demographics are summarized in Table 3.2. Optionally, survey1 respondents could also indicate if they wanted to participate in a follow-up interview.

INTERVIEWS: We carried out semi-structured interviews to probe further into students' experiences with learning programming in an emergency-remote course. We asked them about their preferences for learning resources and strategies, perceived usefulness and pros/cons of internal and external resources, reflections on how they monitored their time and progress, and how they sought help. The goal was to understand how and when students used resources to get help and learn.

We conducted 13 interviews (8F/5M) in group 1. We offered a $15 Amazon gift card to each participant. This survey took 45 min on average via the *Zoom* video conferencing tool. With permission of participants, the interview was audio recorded. Because the data from the interviews was not linked to the responses from survey1, we collected interviewees

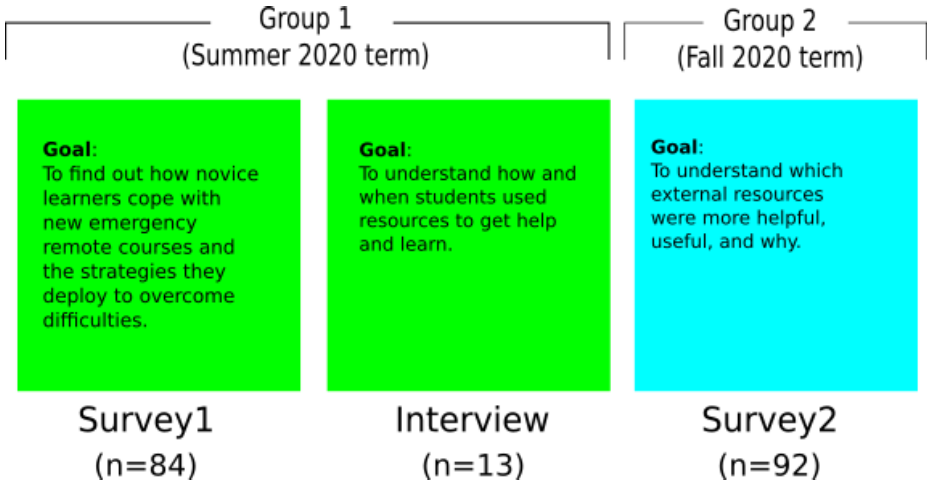

Figure 3.1: Summary of the process we followed to further investigate the prevalence of insights that we saw emerging along the instruments we used.

demographic data via a questionnaire. Demographics from this interview are summarized in Table 3.3.

SURVEY2: After analyzing data from survey1 and the interviews, we wanted to further investigate the prevalence of insights that we saw emerging, especially related to how students were relying on external resources and managing their time (fig. 3.1). Building on our questionnaire from survey1, we asked additional questions about strategies for locating external resources, usefulness of different resource types, and challenges in accessing and learning from external resources. There were 26 questions (23 closed-ended and 3 open-ended). The goal was to understand which external resources were more helpful, useful, and why.

We conducted survey2 in research group 2, from weeks 8 to 12. The survey responses were anonymous. We only offered a chance for $50 Amazon gift card raffle for participation this time. We obtained 92 completed surveys out of 669 (13.8% response rate of the fall 2020 class). Demographics are summarized in table 3.2.

3.3 Analysis and presentation of results

For the closed-ended questions in survey1 and survey2, we looked for general trends using descriptive statistics. We provide charts to clarify our most important findings. We analyzed the transcripts from the interviews and the open-ended survey questions using the NVIVO qualitative analysis software.

For the closed-ended questions in survey1 and survey2, we used descriptive statistics. We started by importing all data to a Microsoft Excel spreadsheet. Then, we eliminated the incomplete rows of data. We considered a row of data as incomplete if the participant did not go through all the questions. We had a total of 5 incomplete rows in survey1 and

Table 3.2: Demographics data from our participants, summarized

	Survey1	**Interviews**	**Survey2**
Term	Summer 2020	Summer 2020	Fall 2020
N	84	13	92
Male	50.0%	5	57.1%
Female	47.6%	8	40.7
Other/Non disclosed	2.4%	0	1.1%
CS Major	9.5%	0	44.9%
Non-CS Major	90.5%	13	55.1%
Response rate of the class	42.0%	6.5%	13.8%

1 incomplete row in survey2. To create our statistical charts, we processed every question individually, considering only the participants that answered that specific question.

Next, we processed the open-ended questions in survey1, survey2, and the interview. To identify recurring themes, we used the grounded theory method [92] by Corbin and Strauss. The grounded theory method is a bottom-up inductive analysis approach for analyzing qualitative data. With this method, we can derive new theories and concepts based on qualitative data, in contrast to other methods in which we can start using an existing theory and see whether or not our data agrees with that theory. The first step of the grounded theory is open coding. In this step, we analyzed carefully the open-answers from interviews transcripts and open-ended questions from surveys, and then broke them up into discrete parts. Later, we proceeded with axial coding, which is the part when we drew connections between the codes we observed in the previous step. Finally, selective coding is the process when we selected one central category that connected all the codes from our analysis and captured the essence of our findings. This coding process was conducted by the thesis author with regular discussions with the other members of the research team. All of the findings we obtained emerged from our inductive analysis and were not defined a priori.

Since our surveys and interviews produced a large amount of data, we combined the key insights into major themes, which are a) to what extent the students' learning process was intertwined with external resources, b) how students looked up for resources in a very quick and shallow way, c) the process students used to assess the quality of the external resources, d) students' perceptions of wasted time and effort looking up external resources and e) if students didn't find what they looked for, whether or not they asked for help. We are presenting our findings in the results chapter.

Table 3.3: Demographics data from our interviewees, detailed

Interviewee	**Age**	**Years enrolled in university**	**Gender**	**Programming experience (in hours)**	**Major**
P1	22-25	More than 5	Female	None	Life sciences
P2	22-25	4 years	Male	None	Business
P3	30+	5 years	Male	None	Business
P4	18-19	Just enrolled	Female	None	Earth sciences
P5	18-19	2 years	Female	None	Life sciences
P6	18-19	1 year	Female	None	Life sciences
P7	30+	More than 5	Male	100+	(Undisclosed)
P8	22-25	4 years	Female	None	Communications
P9	26-30	More than 5	Female	None	Life sciences
P10	22-25	4 years	Female	Between 5 and 30	Business
P11	18-19	3 years	Female	None	Life sciences
P12	20-21	2 years	Male	Less than 5	Business
P13	18-19	1 year	Male	Between 5 and 30	Business

Chapter 4

Results

In this chapter, we report the main findings from survey1, our follow-up interviews, and survey2. Our key themes indicate that although students appreciated the curated resources offered to them as part of their course and they had several opportunities to get one-on-one help, many of them stubbornly believed that it would be more efficient to consult external resources on their own. In fact, their learning was so intertwined with external resources that many would spend several hours struggling with trial-and-error strategies on the web, only to later realize that they were actually wasting their time and effort. Students reported that they tried to obtain external resources by sifting through them in a quick shallow way. This process made students less prone to reading, learning, and reflecting properly. Students also reported they preferred minimizing the time invested trying to solve the exercises on their own, preferring to use someone else's code. They reported preferring not to ask ask for help, keeping this process in a loop over and over again.

We grouped the main findings into five sections that shed light on: 1) how students' learning processes are intertwined with external resources, 2) how students look up resources through quick trial-and-error, 3) students' perceptions of wasted time and effort looking up for external resources, 4) the process students use to assess the quality of external resources and 5) students' reluctance in seeking help from others.

4.1 Students' learning is intertwined with external resources

Table 3.1 shows the list of internal and external resources that commonly came up in our study. In survey1, the respondents reported that they regularly consult the slides provided by their instructor or their own course notes (57.1%). The course textbook was a less preferable choice (23.8%). Most of the respondents also indicated that they accessed other resources curated by the instructor for supplementing course activities and assignments. Almost all of our interviewees (10 out of 13) confirmed that the provided internal resources (course notes, slides, official interactive textbook, course-based forums) were considered useful for them and that "*teaching and the materials provided were pretty good*" (P13). However, we also

observed in survey1 that almost all respondents reported that they ended-up using external resources.

In survey1, we offered a set of five internal and four external resources (Table 3.1) and asked students to select the top three resources they used. We observed that 98.0% of survey1 respondents selected a combination that includes at least one external resource (Fig. 4.1). From the interviews, we learned that the majority of students reported using external resources (12 out of 13) as well. P1, a female student who had finished her major in life sciences and was thinking about going for a second degree in CS, explained the apparent contradiction between having good resources provided by the instructor, but at the same time, relying on external resources:

> *Generally, all of the material that we've done so far, it could be found in the lecture notes or the interactive textbook, Runestone. Generally, those two are enough. It's only when it's something, like a creative assignment or content in the mid-term, that those two I require to use something like YouTube or Google...* (P1).

We found that many of the survey1 respondents (16.7%) even preferred to use external resources as their first choice for studying or working on assignments. Some of our interviewees confirmed this preference for external resources to their own notes or instructor-provided resources. For example, P13, a first-year male student working towards a business major, explained:

> *I think my notes are too way more unorganized than what I could find online. Online is just tabs of exactly what you need...* (P13)

In survey2, we further probed into this finding and learned that the majority of respondents (70.6%) again reported that they relied on external resources. The most common reported reason for consulting these resources was to find something *"exact"* which was difficult to find within the course textbook or other internal resources. P10, a fourth-year female student studying a major in business, explained to us:

> *It is just convenience. I don't want to look through the textbook if I don't know if my answer's there, so I just Google the exact question, and then chances are it comes up... I rarely look at the textbook. Actually, now I think about it, it's a really good textbook...* (P10).

P12, a second-year male student majoring in business, described the *"burden of reading long texts"* compared to the quick hands-on solutions he could get from reading and using someone's code in a Q&A forum, such as StackOverflow:

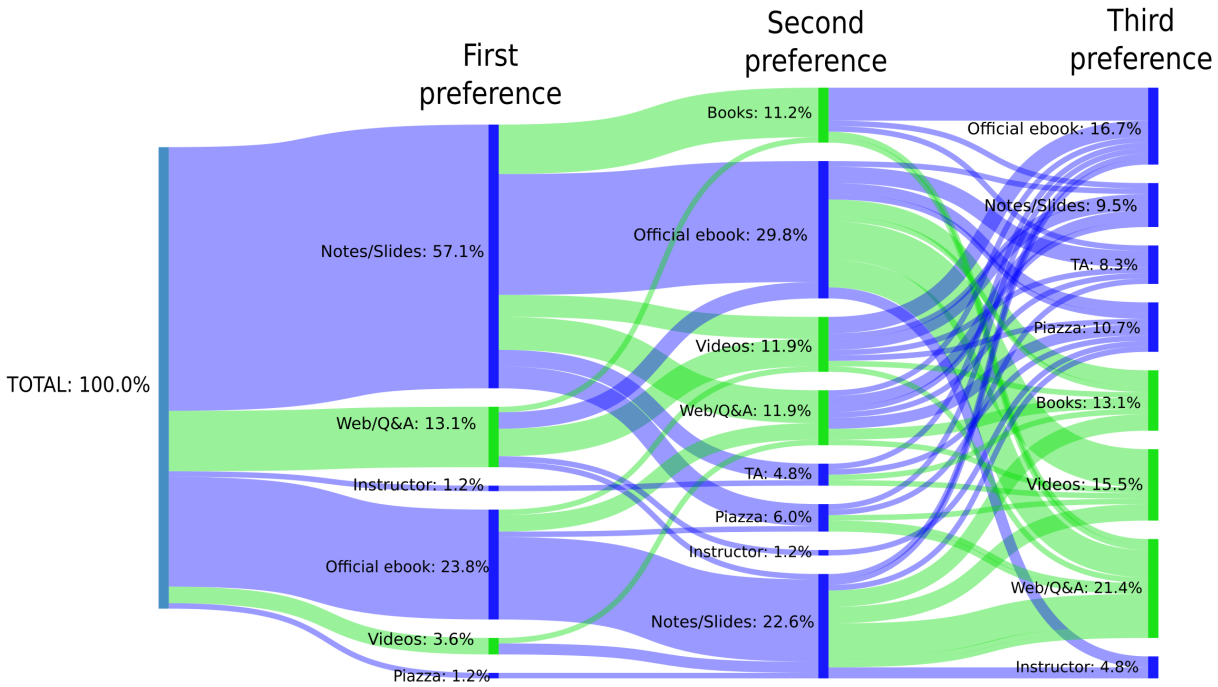

Figure 4.1: Survey1: This graph represents students' preferences for resources. We asked survey1 participants to rank their first, second and third preferred resources when studying for this introductory programming course (from a list of potential internal and external resources). Blue color represents internal resources, green color represents external resources. It's interesting to note how students' learning efforts were so intertwined with external resources. In fact, 98.0% of respondents indicated they would prefer at least one external resource as one of their 3 options (only 2 people said they would stick with internal resources as their first 3 options). Although the majority of participants preferred to start with internal resources (83.3%), an important number of students (16.7%) indicated that they would choose an external resource as their first preferred resource, skipping the provided ones (first column: Web Q&A sites 13.1% plus Videos 3.6%, total 16.7%).

> *I find it easy to look at someone else's code and then take that method and use it for myself. So I don't read the interactive textbook because it's too many words... I'm not a huge fan of reading. If StackOverflow makes it in three lines so you can make it in three lines too...* (P12).

We found that our interviewees reported using external resources mainly to finish their assignments, and that they indicated they were not shy about consulting extra resources. That was the case of P9, a female student who was a recent graduate from Kinesiology and took this course out of curiosity. She explained to us:

> *When I do look out [online], yes, it takes a lot of time, because a lot of the questions that are even being asked on Q&A websites, they are not the exact*

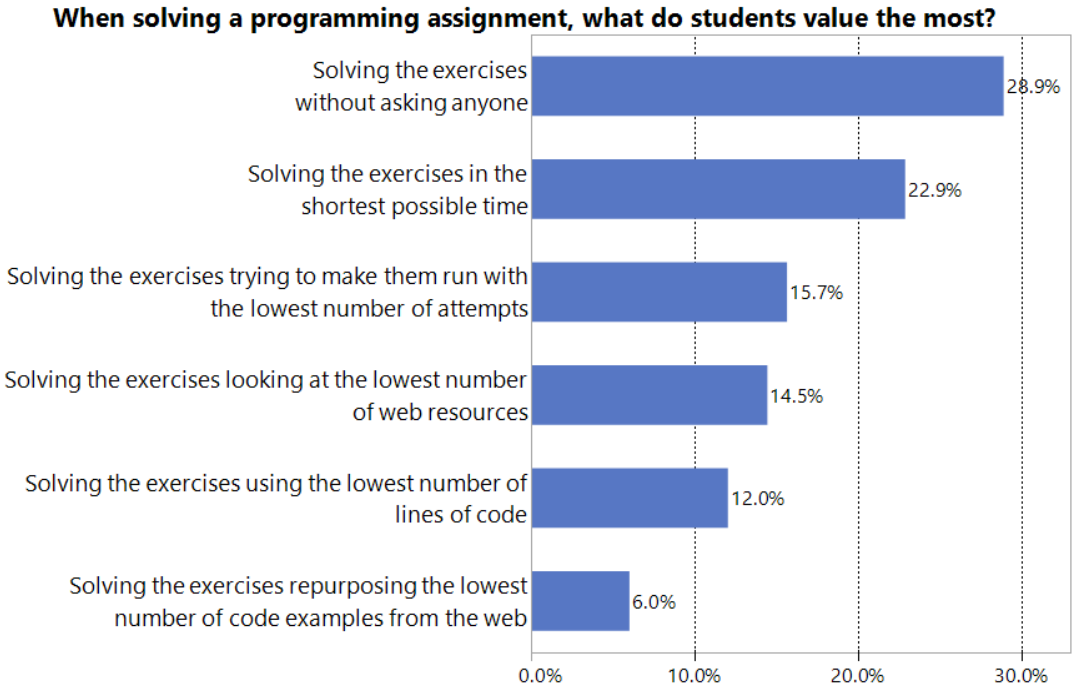

Figure 4.2: Survey2: Respondents indicated that the two things they value the most were: working independently without asking anyone (28.9%) and solving the exercises in the shortest possible time (22.9%). It is interesting to note that most respondents were very open to repurposing examples they find online (only 6.0% mentioned that they would use the lowest number of code examples from the web).

> *solution you are looking for anymore. And if it's something very specific, it doesn't really cover it either. But it does give you a good general idea though, which at least launches you, starts you off somewhere, guides you somewhat in a general direction, I think...* (P9)

In survey2, we asked our participants what they valued the most when solving programming assignments. Their answers are summarized in Fig. 4.2. Only 14.5% of participants considered it was important for them to look at the lowest amount of external resources to solve the assignments; and only 6.0% considered important minimizing the amount of programming examples they repurposed to solve the assignments.

Interestingly, some students even preferred to completely ignore all of the internal resources offered to them by the course instructor. As noted in survey1, we learned that a number of respondents (16.7%) reported starting to look for external resources even *before* opening their own notes, course slides or course textbook (Fig. 4.1). Our second survey confirmed this as we found that 15.5% of respondents also reported preferring to skip internal resources altogether. In the interviews, 2 out of 13 respondents indicated they didn't

use any internal resources at all, further clarifying that they were not enthusiastic about internal resources. P9 explained why she did not rely on internal resources:

> *The textbook doesn't really provide that many examples. They do provide some examples, which could start us off. But I find that there are some problems it just doesn't cover. There are certain questions our instructor gives, and then I'm like, How do you solve them? And then, I go back to the textbook, and it doesn't really have enough, I guess, substance to provide every scenario in the world, which is understandable. You can only have so much in a textbook...* (P9).

We probed into this behavior in survey2 through an open-ended question about students' motivations for using external resources. One of the key reasons that consistently emerged for using external resources was that respondents wanted to get *hints and tips, explore different methods or find ideas for inspiration* (44.1%, Table 4.1). As explained by one of our interviewees: "*sometimes when I code, I don't know how to start, and that's usually my biggest problem* (P11)". Another key reason was that students indicated they perceived that *their prior knowledge was perhaps insufficient* (38.7%) and they told us they wanted to use external resources to close this gap. Surprisingly, a large number of respondents (17.2%) indicated they believed in the power of Google and were confident that they would just be able to get their desired answers when needed: "*the other day I wanted to learn how to convert decimal to binary so then I'll Google just 'converting decimal to binary'* (P8)".

Even in cases when interviewees knew that they could resolve a Python programming error output based on the information in the course slides or their textbook, they indicated they preferred to do a Google search as it *"is just very convenient"* for locating information. P5, a second-year female student studying a life sciences major, explained how she used Google:

> *If I don't know where to start, I'll usually search for that specific function [that was taught]. If I got an error for some sort of syntax error or something, I would try to search whatever that I'm not sure about that might've caused the problem. Or I try to search the different kinds of people who have similar programs that want to do something and maybe not the same word for word, but if their program wants to add numbers and I want to subtract numbers...* (P5)

From the previous quote, we can observe the different reasons why P5 may search on Google: i) to debug code when a syntax code error arises ("*if I got an error...*"); ii) to get unstuck ("*if I don't know where to start...*"); iii) to check different methods or ideas to achieve a solution ("*different kinds of people who have similar programs that want to do something...*").

But, what if P5 didn't get a specific answer from external resources?

Table 4.1: Survey2: Codes and categories from students' answers to the open-ended question: *in which cases do you rely on external resources available on the web? (beyond your own course notes or resources that the course instructor recommends).*

Codes	**%**	**Category**	**% Total**
Check extra topics not covered in class (modules, concepts, etc)	17.2%	When I want to get hints, explore different methods or find ideas for inspiration.	44.1%
Check different methods or ideas for achiving a solution.	14.0%		
When I am stuck (get hints, course of action or ideas).	10.8%		
When no one can help me with my question.	2.2%		
Incomplete explanation (theory) or basic previous material that they don't know.	25.8%	When the perceived prior knowledge or course resources are deemed insuficient.	38.7%
Few examples provided by the course material or lack of variation on them.	11.8%		
When students want to get extra practice.	1.1%		
Seek explanation for code errors, functions or programming syntax.	11.8%	When Google is perceived as a convenient book index.	17.2%
It is more convenient look into Google than sift over tons of materials.	5.4%		

> *If there's no website that has my specific answer, then I'll look for something similar that I can kind of compare my code to and see where I've gone wrong or what I need to look at in my code...* (P5)

From the previous quote, we can observe that although she didn't find an specific answer, then she still prefered to keep looking for more resources in order to find *"something similar"* to *"compare my code and see where I've gone wrong"*. Thus, it is not surprising that resources in which different people offer programming solutions with example code, such as Q&A forum resources, were popular among our interviewees.

In fact, from our interviews, we noticed that students preferred Q&A forums, such as StackOverflow (9 out of 13) and about half (6 out of 13) reported using them as their first choice. Videos, mainly from YouTube, were reported as the next preferred resource (7 out of 13), used by three students as first choice. Survey2 also confirmed this observation (Fig. 4.3) that more respondents (44.0%) stated they found Q&A sites to be very helpful or extremely helpful compared to videos (29.0%). From P3, who was a mature student majoring in business, explained further how he used Q&A websites:

So when you look for something online, it's not per level. It doesn't fit exactly what you're looking for. But it gets a chunk. So I have to get maybe a chunk from this website, and then go to this website, combine it. This course is still entry level, so I would say I'm very beginner in coding. So when I go on the website, sometimes it's intermediate, sometimes it's expert. And I'm pooling a lot of bits of code here and there together, to get to my goal of what I need to research... (P3)

When we asked about videos, this student confessed the reasons why he didn't like using them:

Every time I do my searches, it's always programming forms, or programming websites [that appear]. When I do my search, not a lot of videos come up (...) I didn't think about going to YouTube to search, because let's say I search a topic, it's always broad... (P3)

As P3 mentioned, not a lot of videos appeared in his results page, and the answers he obtained from videos were very broad. However, these were not the only reasons why he didn't like using videos:

Another thing I don't use YouTube for learning this course, because my opinion is that people post that stuff on YouTube to make money. And sometimes they have a title in the video that's not related to what they teach in the video. So unless they have a very specific title, and I can see that it's a very specific channel aimed at teaching Python, I wouldn't go... I would say that I watch maybe less than 5% of my time on videos... (P3)

All of the interviewees mentioned that they used some general webpages (such as Python documentation), but only (3 out of 13) were able to recall a specific web page, and none of them used webpages as a first resource.

In summary, from our interviewees' answers we noticed the following main reasons for their extensive use of external resources. a) interviewees found it more convenient to search on Google than reading long texts; b) they aimed to avoid the frustration of looking for an answer on their own within the textbook, since they reported that they did not necessarily find the exact answer in it; c) interviewees reported that the internet has more hands-on code examples and explanations in different formats which sometimes may be close to what they exactly need to solve their problem; d) the source code or answers from external resources were reported to be good, optimal and easy to build on; interviewees indicated that they did not want to *"reinvent the wheel"*; and e) interviewees reported being convinced that they are skilled in getting what they need from the web. This was observed especially in interviewees who avoided using any course provided resources.

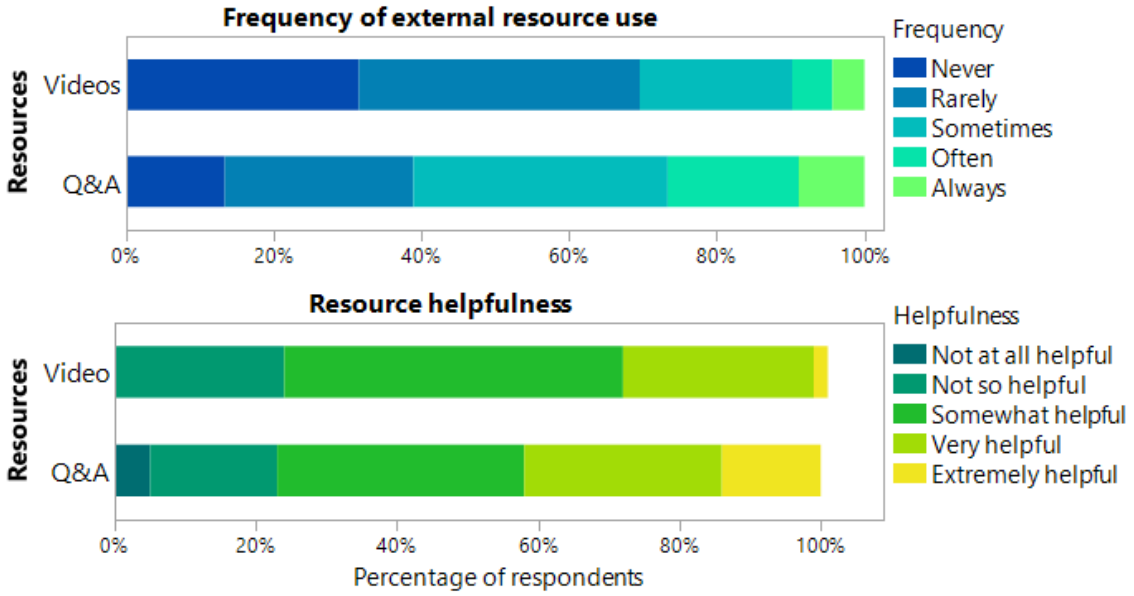

Figure 4.3: Survey2: External resources frequency of use and usefulness. Q&A forums were the most used resource and perceived as the most useful. More respondents indicated they always use Q&A forums over videos (8.9% vs 4.3%). More respondents found Q&A videos extremely helpful compared to videos (14.0% vs 2.0%).

4.2 Looking up resources through quick trial-and-error

Since so many participants in our study were dependent on external resources as part of their learning, we wondered how they may be locating such resources. Our interviewees reported that they mostly rely on different combinations of search queries and trial-and-error.

From the answers in survey2, we observed that the most common reported way to look for resources was by doing simple searches on Google: *"how to do (something) in Python"* (44.5%). We asked our interviewees about their strategies for obtaining new resources and about half of the interviewees (6 out of 13) said that they *copy/paste the exact question into Google*. Notice that the term *exact question* appeared in their answers. A common sentiment among our interviewees was that:

> *Chances are I can just Google it, and I'll find a very similar syntax for it, that someone else has already done it before... because it is a basic course, the odds that the answer is out there seem high...* (P10).

Other interviewees were fairly confident about their ability to find any desired resource online:

> *Internet has all the knowledge. It is hard to fail looking for resources...* (P12).

All of our study participants were part of this introductory programming course, and the majority were learning programming for the first time. They indicated they did not have the appropriate vocabulary and struggled to identify what they were looking for:

> *The problem in terms of coding is, I don't know what the algorithm is. I don't know what I'm supposed to tell the computer to do...* (P3).

P11, a third-year female student majoring in life sciences with "zero programming experience", commented what she did for obtain new resources:

> *What I do is sometimes I copy/paste the assignment into Google. But then, other times, that obviously doesn't work because you can't find every assignment on Google...* (P11).

For such interviewees, if the exact question did not appear in the search results, they would have problems recognizing the answer.

Since students are usually under time pressure to finish an assignment or study for an exam, some of the interviewees described the feeling of being in a *trap* as sifting through search results and resources quickly, they were only getting a shallow insight into the key concepts, not learning properly, which kept reinforcing this trial-and-error loop. Most of the time, our interviewees reported being in a continual *scanning mode* and rarely tried to read and understand the content properly.

In addition to being pressed for time, the students in our study also did not have the appropriate vocabulary to understand complex explanations. Most of our participants indicated that they used Q&A forums more frequently (since they had fewer complex explanations) and found them to be most useful (from survey2, Fig. 4.3). Study participants reported that the key reason to use Q&A forums was that it was easier for them to get an *exact match* for their queries.

From survey2, we probed into attributes of Q&A forums that participants preferred. We asked our participants what they valued the most when looking for answers in a Q&A forum. We learned that the most rated attribute for Q&A forums was that the question text in the Q&A post should be similar to the *"question I have right now"* (37.5%, Fig. 4.4). One of our interviewees, P12, who was a frequent user of external resources (and even skipped any internal resources), explained this common behavior among students preferring Q&A forums:

> *If it [a forum post] answers my question, then I use what the code has taught me (in Q&A post) and then I apply it, that's the number one. Generally, I just use the resource, then like I'll skim through it at first to get an idea of whether or not to answer the question, and if it really does, then I'll actually properly read it to understand what it's trying to tell me and if I can understand it, I'll be good...* (P12)

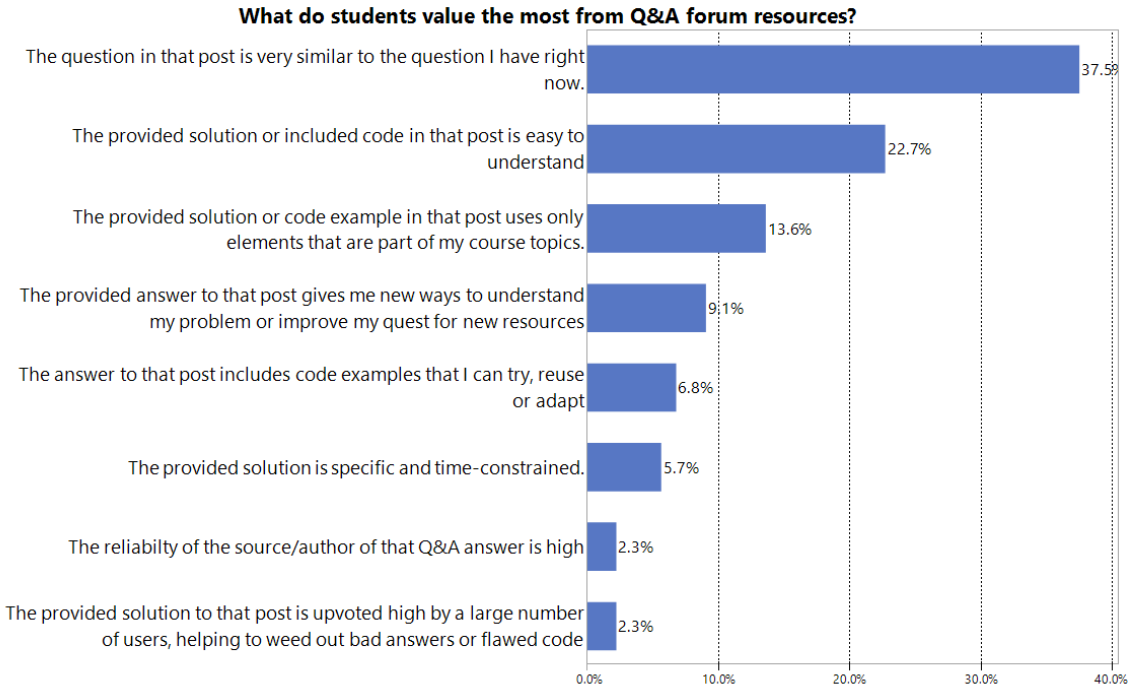

Figure 4.4: Survey2: What respondents valued the most when looking for answers in a Q&A forum resource. The most important thing that this survey respondents (37.5%) valued was that *the question in that post is very similar to the question that I have right now*. Interesting to note that the reliability of the source/author and whether or not the post was upvoted by the community were marked as the lowest attributes (2.3%).

In the interviews, we also learned that some of the reasons students reported Q&A forums as being popular were that they offered different solutions or courses of actions for a problem, often including source code (5 out of 13). These different solutions helped students to get inspiration or ideas to start, especially when they were stuck and did not know what to look for: "*if I had a problem and I'm unable to find a code which at least solves part of the problem, if not the entire part, then I would look at other codes* (P4)". However, the major drawback that students noted for Q&A forums was that it was hard to understand *"other people's code"*, especially since it was often *"beyond the level"* of the student's current knowledge or that there were *"too many different solutions"* from which to choose.

Another interesting dilemma that students reported facing when using someone else's code from Q&A resources was how to adapt these external codes to mimic their own coding style or limit the code complexity in order to make credible answers. P12 explained what he valued the most from Q&A resource was the clarity of the provided code:

> *Stack Overflow has answers that are, God knows how long, and then it has so much code and such complex code that you can't apply it to this course. If you*

do, then obviously the professor's going to know that you used those really high level coding, you shouldn't be to here then... (P12)

Thus, it is not surprising that sometimes, some students reported that they could solve assignments without realizing how they solved them. P12 further pointed out:

For Turtle, I don't know how I did it. See, half of the time, I don't even know how I learn these things. It's because I generally look at the problem and then I'll look at many resources and now I'll just create a puzzle in my head of how each code works and for some reason it works... (P12)

Still, the reported strong preference for Q&A forums is surprising because in survey2 participants also indicated that videos provided clear explanations (42.0%, Fig. 4.5). In the interviews, we learned that videos were reported to be more beginner-friendly and accessible, similar to a recorded tutorial (3 out of 13). But, many of the interviewees (7 out of 13) reported several drawback of videos as well as many programming videos were perceived as being *"long"*, *"tedious"*, and *"slow to use"*, and difficult to use to *"obtain specific answers"*.

When a search did not yield desired results, almost all interviewees reported that they would reformulate their queries by making slight tweaks (e.g., by adding or removing words like "*Python*" or "*examples*"). None of them indicated using any advanced search techniques such as using logic operators, quotes in the search bar, or search filters. In fact, 3 of the interviewees revealed that they relied on Google *"suggested words"*, which means that even their search query was partially suggested.

Surprisingly, our interviewees reported neither a single comment about self-monitoring their trial-and-error process nor a comment about some explicit effort to take notes or adjust their overall strategies. When we asked about the difficulties of obtaining new resources, most of them expressed a lot of frustration:

(getting the exact answer) is not perfect every time. I know sometimes, the prof designs questions that are just extremely unique, and they're really not usable in real life... (P10).

Interestingly, even after feeling frustrated or tired, the interviewee admitted that they would just go back to trying the same thing again the next day.

Another common sentiment expressed by interviewees was that obtaining resources was not the issue. Instead, the core issue was deciding which one to use from the *"sea of resources"*:

It may take more time sorting through the resources and finding which one I want to use... finding one that's easy to understand is probably what takes the bulk amount of time... (P5).

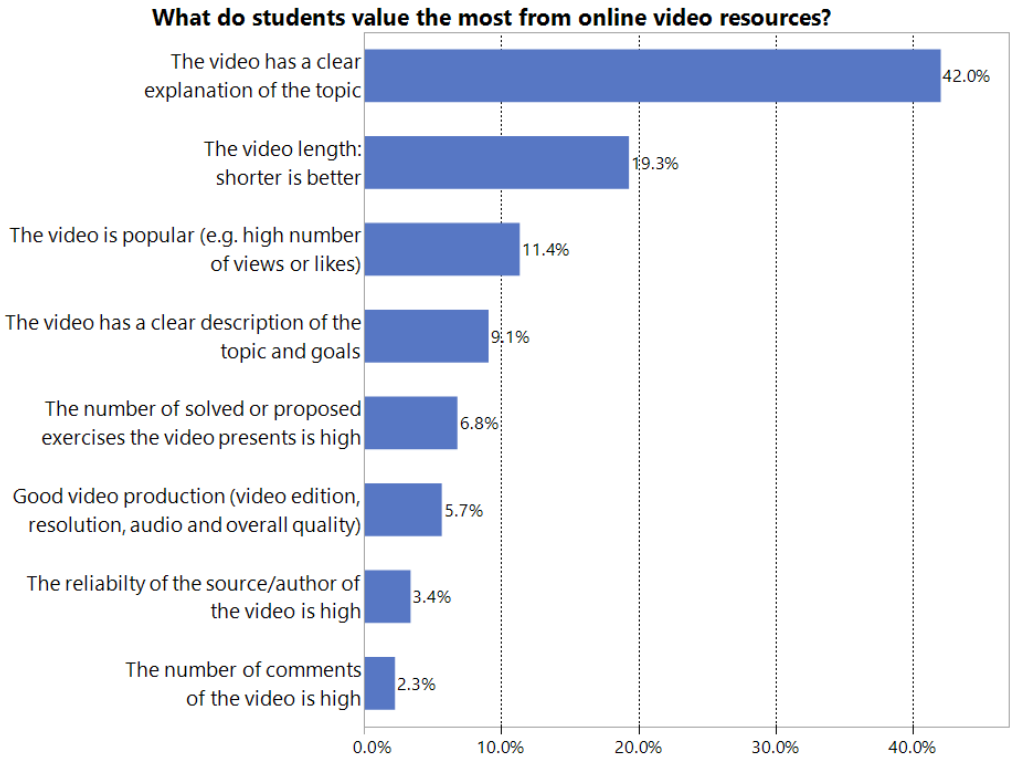

Figure 4.5: Survey2: What respondents valued the most when looking for online video. The most valued aspect of selecting videos (42.0%) was whether *the video has a clear explanation of the topic*. Similar to the Q&A forums, it is interesting to note that the reliability of the source/author was not highly valued (3.4%).

Like P5, many students hinted that they were ready to jump into the next resource at the first problem or confusion. Their comments rarely reflected on their own ability to understand the concepts and blamed the resource for *"not being clear"* or *"not easy enough"*, made a new search and kept looking, making the wheel of this trial and error approach spin.

4.3 The process students use to assess the quality of external resources

An interesting observation from this study is that even though the novice learners did not yet have the skills or vocabulary to assess external programming-related resources, they still believed that they could determine the quality of such resources on their own. This seems to be a paradox because they might profit more from using external resources of proven good quality. This sentiment was general along our research tools: Only 2.5% of survey1 participants indicated that it is important that an external resource comes from a well-known or high-quality source (speaker, publisher, top web site, etc). In the interview, we found that just one interviewee indicated that he followed some good-quality YouTube

channels, but then this student realized that he didn't use videos from that channel very often because that channel was unlikely to answer all of his questions, so he needed to look at other channels suggested by YouTube.

All of the interviewees explained that to assess the quality of a resource, they looked for certain details. These included, for example, the video description, the web page layout, user interface and navigation, quality of the visual elements, number of likes or comments, and the presence of Python code. One interviewee explained:

> *I like to look at what the website is like, first of all. I guess there isn't really a way to make sure that it's completely reliable, but if it looks like it is a little spotty, I might take a sample of the code and put it into my code and just write it and make sure it actually works. And then I can kind of go, oh, okay, well it works, so I'll look more into it...* (P5).

P13, a first-year male student from business major, used a different approach to evaluate the quality of the resources he found on the web: He told us he compared external resources with one another to see if they agreed. He explained:

> *So then, I search it [the topic] up not just on one website but on multiple links that it comes up and if I see that the same thing pops up, if the same answer comes up from those different websites then I think to myself, oh, that must be how I'll do it because that's how multiple people are solving the problem...* (P13)

The problem with that approach is that it takes much longer compared to just relying on external resources from a proven good-quality source.

We found the same sentiment about disregarding the author and quality of the external resource among survey2 respondents. We asked questions about what students value the most in the case of webpages, videos, and Q&A external resources. The percentage of students that selected the reliability of the source/author of the resource as a main reason was: 4.3% for web pages (the least desired attribute for web pages), 3.4% for online videos (the second-least desired attribute for videos), and 2.3% for Q&A forum resources (the least desired attribute for Q&A resources). The grounds for this behavior may be that students have difficulty in recognizing reliable resources and struggle in even knowing what a reliable resource looks like. Students might prefer looking for new external resources from whatever place is offered by the results page, instead of just relying on a small set of specific places of good quality. They also seem not to worry too much about the reliability of the source, but about the development of a way to assess the resource quality on their own. For that, they looked for visual cues, analyzing the information layout, copying and pasting the provided code to see if that code runs, or comparing the resources they found with one another. All of those options are time consuming for such novice learners of programming.

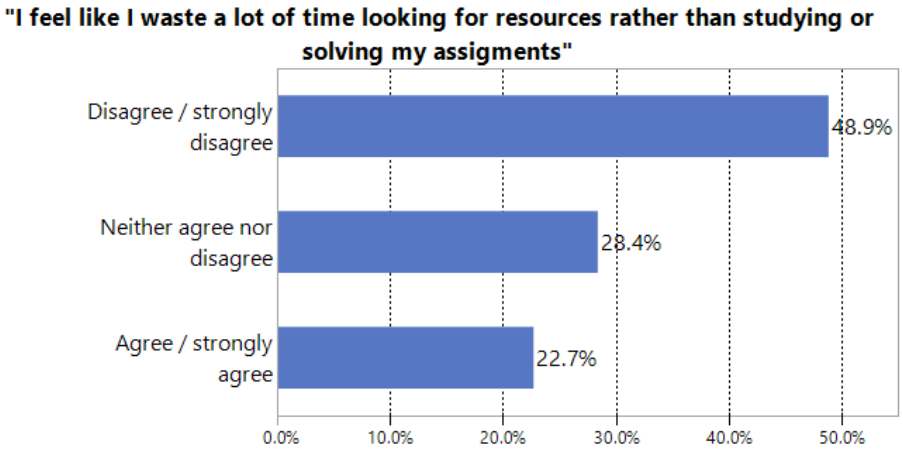

Figure 4.6: Survey2: Based on our interviewees' responses, in survey2 we went deeper into students' perceptions of wasted time when looking for resources. We expected that most students would not consider their online searches to be a waste of time. But, surprisingly, we found that only 48.9% of respondents in survey2 disagreed that they were wasting time. In fact, about a quarter of respondents (22.7%) strongly agreed or agreed that they were indeed wasting their time. Yet, as we note in our results, these students reported making little effort to change their strategies and be more efficient with their time.

4.4 Students' perceptions of wasted time and effort looking up external resources

We observed that many participants consistently expressed the sentiment that the *"internet has all the knowledge"* and that is why they kept pursuing external resources. But, survey2 showed that a key desired goal for the study participants (22.9%, Fig. 4.2) was to be able to solve their assignments in the shortest possible time. So, it was ironic to hear them spending so much time and effort going back-and-forth to locate relevant resources and being unwilling to change their strategies even when they were unsuccessful.

To further our understanding of the findings from survey1, we explicitly asked participants in survey2 to what extent they agreed with the idea that they were wasting time looking for resources rather than studying or solving their assignments. We expected that most students would not consider their online searches to be a waste of time. But, surprisingly, we found that only 48.9% of respondents disagreed that they were wasting time. In fact, about a quarter of respondents (22.7%) strongly agreed or agreed that they were indeed wasting their time (Fig. 4.6). Moreover, 11 out of 13 interviewees agreed that looking for external resources was a "time-consuming process". As one interviewee explained:

> *I apply what I learned from the video or website to my work...it does take sometimes even a full day. So I do admit that. I do waste a lot of time doing that...* (P11).

Further quotes support the idea that it was hard for interviewees to control the time they spend looking for resources. For example, only three of the interviewees mentioned that they had some awareness of time when they were searching the web; others commented that they relied on internal cues to stop their search (e.g., feelings of frustration, boredom, or fatigue). P8, a fourth-year female student studying a major in communications, explained:

> *It's pretty easy to realize how much time I am spending on external resources, because the longer that I take the more frustrated that I get so, the more frustrated that I am, that indicates to me that I'm taking a long time where I'm taking too much time looking for resources...* (P8)

P12, one out of two interviewees who skipped using any internal resources, told us that the idea of measuring time was "nonsensical" for him:

> *It shouldn't matter if you spend two days on looking at external resources or one minute, if it's towards answering a specific problem and you need to find the answer, it can never be a waste of time...* (P12).

Another common behavior that appeared in our study was that some students reported trying consistently to minimize the time they spend on each resource, so that they could maximize the number of *"unique hits"* and increase the chances of getting the *"right resource"*. As expressed in the previous excerpt from P12, *"find the answer"* and *"specific problem"* were all common phrases used by our interviewees, especially in relation to how they used Q&A forums. All of them mentioned they wanted specific answers, especially when they were up against deadlines and sifting large amounts of resources was a difficult task. In that situation, they could not afford to spend time learning the content properly, as a result.

In addition, some students reported that when they felt that a deadline was approaching, they would resort to using even more external resources, which would ironically add more time. Some students admitted how they tackled this *"fight against time"* by taking ideas as-is from the resources, without thinking too much:

> *If I'm running out of time for the assignment it's best to just, oh, this is how this person did it, that must be right. I'll do the same thing. And then it kind of takes away a lot of stuff in between. It cuts down the steps...* (P8).

Given the time crunch that some study participants indicated they faced, it was not surprising that these participants reported they were focused on locating resources where the code output matched their own desired output or something similar to get ideas or inspiration. They even commented that they felt that all the time and effort was *"worth it"*

when they found the desired information. However, the main issue they hinted they faced was not having the confidence to know when to stop, even when they realized that their time was not being well-spent:

> *Sometimes I feel I'm wasting time (looking for resources), at the end of the day, I have to find the materials to look at... if I don't get the materials, how would I approach my assignment?* (P13).

4.5 Reluctance to seek help from others

Since interviewees reported relying so much on external resources and perceiving that their time and effort may not yield useful results, we wondered if they sought help from others to get unstuck and break away from the harmful trial and error cycle. However, we observed that interviewees reported remarkable reluctance to seek help from others, especially from the course instructor and TAs. This was also ironic because students admitted that in the case of online courses, it was much easier to book online appointments compared to face-to-face courses. From the students' perspective, *"instant answers"* that they could find on their own were reported as more efficient and made them feel that their learning process was under control. On the other hand, asking for help was consistently reported to be the opposite: slow, inaccessible, and potentially high in terms of social costs.

In survey2, participants indicated that solving a programming assignment *without asking anyone* was the most valued attribute (28.9% of total, Fig. 4.2). This corroborates the sentiments expressed by most of our interviewees (8 out of 13) who said that asking someone for help would be the *"last resort"* or the *"least desirable option"* reserved for situations when everything else had failed. We also noticed that none of the interviewees mentioned posting questions in external Q&A forums or in the YouTube comments section: they indicated they preferred reading someone else's questions and answers.

Since all of our interviewees were from non-CS majors, they hinted that they were less interested in programming and felt that they were not confident about asking questions. Only two 2 out of 13 interviewees indicated that they would post a question on the Piazza forum (an internal forum with course peers set up by the instructor), and only 4 out of 13 conceded they would ask something from the TAs or the course instructor. These students reported that they perceived the instructor's or TAs' help would be limited to conceptual explanations, just hints, and would be only for a short period of time given the large size of the class. Students indicated that they would prefer to get a complete explanation or some discussion with code examples (similar to posts from forums such as *StackOverflow*), but asking the instructor for this level of guidance would be inappropriate. P6, a first-year female student from life sciences major, explained to us:

> *It would be really difficult for me to go to the instructor and ask them, because this is a problem, this is the homework. Why am I asking the instructor?...* (P6)

Some interviewees explained that they would also be aware of and ashamed by the number of doubts or (*"little problems"*) they had, and even by how to frame their questions. Therefore, they would only seek help for *"bigger"* issues:

> *Yes, I do waste a lot of time looking for resources, and I think I justify that because I don't really want to go to the TA (teacher assistant) or the prof with every little problem that I have. So I really like to save going to them for big problems that I absolutely cannot solve or no help at all...* (P11).

This perceived social cost of asking the instructor or TAs for help also appeared when interviewees commented about making some questions to their friends:

> *(I don't like to bother my friends) because it's not their job to teach me. So then I try to take what's given to me first and then if I've exhausted all the prior resources, then I go to them...* (P8).

Some interviewees expressed their challenges in even knowing whether or not their problem was *"big enough"* so that it would be worth obtaining help. Their instinct was to be *"absolutely sure"* that they had exhausted all the alternatives to get answers from external resources first. Hence, the cycle of trial-and-error continued incessantly.

4.6 Summary of results

In summary, our findings indicate that the students' learning process was largely intertwined with external resources that they found on their own, mainly forums and videos. We also observed that students reported scanning different resources in a quick and shallow way with the goal of trying to obtain an exact answer. That exact answer meant matching their questions word-by-word, mainly from Q&A website forum resources, although online videos were reported as more beginner-friendly. Because of this shallow process, students reported that they rarely fully read nor reflected on each resource, and kept getting stuck in a loop over and over again. They also did not consider the quality of the external resource on their own and realized later that they were often wasting their time. Finally, when students were unable to obtain the desired resource, they indicated that they didn't stop to ask for help; instead, they preferred to keep searching on their own to obtain a desired answer.

Chapter 5

Discussion

In this chapter, we briefly reflect back on our results and discuss some possible reasons why students may have followed the strategies that they did in the emergency-remote learning scenario.

In the previous chapter, we presented a detailed case study of students in a emergency-remote introductory programming course and how they perceive and use different learning resources. Through two surveys and a series of interviews, conducted with two different groups of students in consecutive terms, we observed that the majority of students were fairly stubborn in their pursuit of external resources. Most of the time, they were not looking for beginner-friendly or alternative explanations in these resources. Instead, they used specific queries to look up exact matches relying on trial-and-error approaches, relying mostly on Q&A forums (unfortunately, these are mostly created for professional programmers, not students). Although students were aware that that they wasted large amounts of time on trial and error in looking up content, ironically, they were still in a constant *scanning mode* exploring external resources. Most of our participants rarely stopped to reflect nor did they ask for help.

From our study, we observed that students' learning process is deeply intertwined with and heavily reliant on external web resources. In fact, 16.7% students in survey1 and 15.5% students in survey2, reported that they jumped right into external resources, ignoring any provided course resource. We believe that this is a major problem for students because looking for external resources sets a burden on them to obtain *good* learning resources (or at least, perceived as *good* for them) when they're still grasping the basic concepts, thus wasting a lot of time in the process. They hinted trying to obtain these resources following a shallow and quick process, focusing on resources that are specific to solving programming assignments rather than focusing on resources to help them foster comprehension. This behavior was reported to be more acute when they were running short on time.

This emergency-remote learning scenario created a difficult situation for novice learners in programming. In this scenario, students had difficulties to access regular in-person help from peers and instructors. We may think that this situation created some impact on the

demand on online resources in order to obtain help that worth investigate. This thesis shed some light on understanding how students used these online resources to obtain that help. We found that novice programming learners in emergency remote learning relied extensively on online resources to obtain this desired help. It was interesting to see that they resorted to online resources on their own, in line with the previous studies about online learning that found online courses in general tend to reuse online resources [60]. This resource-seeking behavior has recently also been studied in the context of in-person programming courses [57] and our study complements these findings from an emergency-remote context.

One of our main findings was that our study participants reported using Q&A forum resources more widely than online videos. This was surprising given that these Q&A resources were mainly oriented for professional programmers [72] and not for novice learners like the ones we studied. Our study participants acknowledged that videos were more beginner-friendly and better suited for novice students, but they still preferred using resources such as Q&A forum resources because that they were easy to be scanned. This need for resources that are easy to be scanned has been found in other online learning settings such as MOOCs [33]. We know from previous studies that when people face difficulties to understand something, they prefer to issue shorter queries to Google [102]. Q&A resources could be seen as short questions with answers, and surely some of our participants faced difficulties in understanding concepts in this introductory programming course.

From prior literature, it was also found that Q&A forum such as StackOverflow may not support the learning needs of new online learners [82]. From this same study, we also know that a main problem when using specific answers from Q&A forums is that specificity on obtaining exact answers discourages users from frequently interacting on these forums [82]. It also means students refrain from discussing with other users, making extra questions or asking for clarifications, thus failing to self-reflect on their approach and learning. In our study, we found that our study participants neither reported they posted any new question on external Q&A forums nor engaged in any existing Q&A forum discussion. That finding from our study is in line with [82]. Our study also contributes to report to what extent novice learners in programming in a university setting fail to self-regulate their approach using these online resources, wasting time and effort but at the same time, being unable to break that loop and ask for help. We know that self-regulation is crucial for succeed in solving programming problems [62]. Thus, this lack of self-regulation when using online resources may have repercussions on their overall learning.

There are several theories that may explain why users don't use the provided course resources and prefer to look for new resources on their own. One of the well-established ideas comes from Pirolli and Card's Information Foraging Theory [75] [76]. This theory compares users' information foraging with animal foraging. Carroll and Rosson's *paradox of the active user* [22] [95], explains why users don't like to read manuals (learning activity) and jump into the production activity. Blackwell's Attention Investment theory [15] explains

how users consider investing attention and time in order to program a task that otherwise would have to be solved manually, wasting more attention time. Williams's [105] definition of gambling provides us with some background to compare students' behavior in terms of gambling. Next, we will discuss these theories in detail.

5.1 Analyzing students' preference for external resources under Pirolli and Card's Information Foraging Theory

Information Foraging is a theory proposed by Pirolli and Card [75] [76] that explains how users forage for information in the same way that animals forage for food. The authors found several similarities, such as a) **GOAL**: the goal of any animal is finding food. In the same way, the goal of any user is obtaining information; b) **PATCH**: a patch is a site containing one or more potential sources of food. Likewise, a website is a potential source of information; c) **FORAGE**: animals forage for food in the same way users forage for information; d) **SCENT**: animals use scent to assess how likely is that a given patch will contain food, in the same way a user uses some cues to assess how promising a potential source of information is; e) **DIET**: animals have a diet, that is the totality of food types that an animal may consider to satisfy its hunger. Similarly, a user considers specific sources from the totality of sources of information in order to satisfy his need for information.

Through this theory, we can explain some of students' preferences for external resources. For example, we can establish similarities in the following way: a) the goal of any student is obtaining promising information to help them overcome difficulties when solving programming assignments; b) the patch of information is a web resource, such as a YouTube video, a StackOverflow post, and so on, mostly found using Google; c) students forage for answers to their questions in the same way an animal forages for food; d) students try to assess a web resource, scanning quickly several sites to see how promising these potential sources of information are in order to invest time in them; e) students' information diet consists mainly of web resources that are easy to find and scan, short in extension, specific to answer one main question and written including source code that can be copied and pasted.

However, we were unable to explain completely the students' preference for external resources using this theory. For example, this theory considers that user behavior when looking for resources is purposeful, but we observed that novice learners' search is not. As we reported in the results chapter, several students from our study group seemed not to know what they were looking for, and they didn't have the experience to identify what they were looking for looked like. Some prior work [74] used information foraging with professional programmers' information diet and found that the main use of external resources for professional programmers was to chase programming bugs, which in was not our case, the main reason for students to look for external resources that we have observed is that they

Figure 5.1: Information Foraging theory [75] [76] explains how users forage for information in the same way this adult cat forages for food through a diet, recognizing its scent (left image). However, first-year introductory programming students lack the experience to adequately forage for information, not being able to recognize their diet. This may make students waste time and resources in the same way this kitten lacks the experience to advert a trap (right image).

wanted to get hints and tips, explore different methods or find ideas for inspiration (44.1% from survey2).

Furthermore, using this theory, we were unable to explain why some students' 'diet' consisted of advanced resources, such as programmer's Q&A answers, instead of videos recognized as better suited for them. Finally, we were unable to apply the *information scent* idea, proposed by the information foraging theory, upon reported students' behavior. This possibly happened due to internet advertisers interest for students' attention time. This differs from what the information theory suggests, making this scent more similar to a bait, taking advantage of students' lack of experience when looking for programming resources (Fig. 5.1).

5.2 Analyzing students' preference for external resources under Carroll's Paradox of Active User

The *Paradox of the Active User* [22] is a concept introduced by Carroll and Rosson in 1987. It tries to explain a very common situation observed several times when users deal with new software: users don't like to read manuals. They just start using the software right away. Their motivation is just to get started and to get their immediate task done: they don't care about the system, and thus don't want to spend time up front getting established, setting up, or going through learning packages. This behavior is called *"production bias"* because users want to start producing instead of learning how to use the system properly. This is a paradox because users would save time in the long term by taking some initial time to

learn how to use the system properly, however, they prefer to jump into production mode, wasting more time in the long term.

Carroll and Rosson's work is relevant because it may help us explain why students in an introductory programming course tend to use external resources. This trend seems more similar to a production activity. Reading the course materials would seem more like a learning activity. Using the concept of Paradox of Active User, we may reflect about, for example, why students don't like reading the provided textbook and prefer jumping right into the hands-on process of using code examples from external resources. They just copy and paste the codes, modify them and learn on the go, just like users tend to skip manuals and prefer to jump into using the software right away. Our belief is that students and users would make their own work more fruitful if they read the provided resources and manuals, respectively. Following the concept of the Paradox of Active User, we can also understand why students prefer reading Q&A posts with source code, instead of watching explanatory videos.

Although this Paradox of Active User may explain why students tend to use external resources, it was hard for us to explain students' behavior completely using this theory. For example, the Paradox of Active User itself can not completely explain the reported students behavior of looking at external resources in a non-purposeful way, not looking for something in particular. The Paradox of Active User does not explain why students keep trying to find some external resources that may not exist. In addition, it could be argued that the production bias for this case is in fact that students program the assignment, instead of jumping into external resources, which might be seen as a learning activity as well. We also noticed that some students seemed that they were trapped in the loop of looking for external resources, neither producing nor learning. This doesn't seem to be explained by the Paradox of Active Users either.

5.3 Analyzing students' preference for external resources under Blackwell's Attention Investment Theory

Alan Blackwell's Attention Investment [15] is a theory that explains how users engaged in a programming behavior ponder the costs and benefits of coding a task vs. solving the task manually. This theory is influenced by the "attention economy" concept, which argues that the scarcest economic resource on the Internet is human attention [40] [78]. Creating a program requires some amount of an investment. The payoff, if the program works correctly, is that it will automate some tasks in the future, thereby saving attentional cost (the user does not need to concentrate on a task that has been successfully automated). There is, however, a risk that the investment will not pay off: there might be bugs in the program, or the task of coding might be way more expensive in terms of attention than solving the task

manually. The Attention Investment Theory describes that users engage in this investment analysis trying to minimize attention cost in order to obtain maximum profit.

This theory may be useful for explaining our students' behavior when using external resources for saving attention time. From the point of view of this theory, students may be aware of the fact that the cost of solving the programming assignment thoroughly on their own, in a polished way, with no copying/pasting, isn't worth the benefit because students may not perceive these programming tasks as something that will save them attention time in the future. Therefore, solving the assignments by taking shortcuts, such as copying/pasting code from external resources, is a good investment for students because it saves them attention time. This also explains why students prefer textual resources that may be easily scanned and offer code that may be copied, such as Q&A forums instead of videos that don't offer these options.

Nevertheless, this theory may not completely explain our students' behavior. For example, this theory was conceived to explain the behavior of both professional and end-user programmers when deciding if it is worth investing time coding a task or solving it manually, which is not the case of novice programming learners, who mostly can't decide whether or not to invest in coding the assignment because these are requirements for their course. Another difficulty is that this theory considers that users should have some experience in programming to properly assess the investment in attention time, experience that novice programming learners lack. Finally, this theory does not explain why students are not able to stop looking for resources when they do not find anything useful, which is a bad investment in terms of attention time.

5.4 Analyzing students' preference for external resources under William's gambling definition

From our study, we observed some similarities between students' behavior when looking for resources and gambling. We believe that students' behavior, when looking for external resources, might be analyzed in terms of gambling. Williams et al [105] define gambling as *"staking money or something of material value on an event having an uncertain outcome in the hope of winning additional money and/or material goods"*. This definition includes five core characteristics, including putting something at stake, seeking a prize, and dealing with an uncertain outcome. We summarize the parallels from our findings in relation to this definition of gambling in (table 5.1) and note some key observations below.

Our data analysis supports the idea that students had a strong desire to obtain a short, exact and relevant answer for their doubts from google searches. This can be seen as a pursuit of the *"main prize"*. In terms of gambling, finding the exact solution is the student's *"jackpot"*. The presence of Google's famous *"I'm feeling lucky"* suggests that there is an element of luck when looking for information [58]. We saw that students' had a similar

Table 5.1: Observed similarities with gambling behavior

Williams's gambling [105] core characteristics	**Observed similarities from our study group**
1) The person is staking something valued that people are willing to pay money for.	The more time students spend searching the web, their screen time [109] and personal data [65] can be potentially exploited by advertisers and internet companies.
2) The purpose of making the stake is to win additional money and/or material goods.	Students stake time and personal data to get a specific answer from the web.
3) The prize consists of something valuable.	The prize is a specific answer for students' assignments, that students deem valuable.
4) The stake is lost or the prize is won depending on the outcome of an specific event that will occur.	The assignments have deadlines at specific times, which could be considered as the "end of the game of findings resources"; at that point, the game is lost or won.
5) The outcome of the event is uncertain.	The terms "being lucky" or "have the chances" suggest that the process of looking for resources is indeed uncertain for students.

perception of web-based resources and believed that they would get "lucky" and obtain their target jackpot as long as they were willing to keep trying. Instead of reading the textbook or reviewing their own notes, students optimized getting more answers in less time. It also explains the students' preference for obtaining short answers from Q&A forum resources over watching explanatory videos, even though students admitted that videos were probably more adequate for learners. Ironically, by searching the web using trial and error, students realized that they were sifting through countless, random pages and wasting time, but like gamblers, wasting time is perhaps the price to play.

Overall, we found that students' lack of self-monitoring strategies makes them resort to a basic *look-up and match*. This suggests that they are relying on recognition and recall, which is the lowest form of learning in Bloom's taxonomy [55] [94]. Even as they lose track of time, they are driven by a strong belief that the jackpot will solve their problem quickly

and effortlessly which keeps them engaged in their pursuit. They prefer searching instead of stopping and reflecting on their gaps and asking for help. If they were to ask for help, they would be slowing down their pursuit of solving the assignment. Thus, the decision of using a simple *look-up and match* on Google was much easier and required no social cost.

We recall from interviewees who skipped all internal resources mentioning that they think that looking for external resources is a *"challenge"*, and they believe that they are very good at defeating it. One of them even explained that on the day we interviewed him, he had one assignment due in the next two hours, and he was confident that he would *figure it out*. This was the same interviewee who told us that sometimes he solved an assignment puzzling from other source codes and "*for some reason it works*" (P12).

From our studies, we also observed that students are aware of their chances to obtain the main prize, displaying some strategies to minimize their loss and maximize their profit. We observed that they started with a bold maneuver, which is copying/pasting the exact question to Google, no matter how long the question is or how unlikely it seems to obtain an answer. With this move, they are trying to maximize their chances to win: Copying/pasting to Google and obtaining an answer from it is almost effortless and takes a few seconds. From that, if they don't obtain something valuable, they would create a strategy breaking-down this big bet on small bets with less profit, but safer in terms of time. For example, students may try to break down a large problem into a few problems whose answer is easy to obtain, trying later to mix and match all the answers from different sources. Given these suggested parallels, we consider that the novice programmer students in emergency-remote courses reported behavior might be characterized as gambling-like behavior.

Chapter 6

Implications for future work

Our study raises important concerns about the high reliance on external resources that we noticed among novice programmers, discussed in the previous chapter. We wonder to what extent it is possible to create some tools to help students avoid the high reliance on external resources, and with it, avoid wasting time. Instead, this time could be used to solve assignments creatively, reflecting, and recovering from coding errors, thus expanding their programming knowledge. In the end, the goal is not just to solve the programming assignment but to build students' strengths, confidence and skills to express a solution using algorithms. With this approach, they would be able to solve a myriad of problems using different programming languages.

Reflecting back on our results, we consider some directions for future work from and HCI and Computing Ed perspective. In particular, we suggest tools and design ideas to overcome students' high reliance on external resources that we observed in our study.

6.1 Future work: Helping learners from an HCI Perspective building new tools

HCI is the area of computer science that deals with human aspects of using computers, helping to improve the understanding between users and computers. The Association of Computing Machinery (ACM) defines HCI as "*a discipline that is concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them*" [46]. An important focus of HCI is how to boost human performance when using computational tools or software. Therefore, supporting learning is a significant part of it. An important area of HCI research is investigating how to help novice programming learners, especially when difficulties or limitations arise [108]. To achieve that, User-Centered Design [70] [69] plays an important role. User-Centered Design is the principle in which users' needs are set at the core of the designing process, involving the human perspective in all steps of the problem-solving process.

Thus, from this HCI's User-Centered Design perspective, we discuss some implications of our findings. We can consider students' reliance on external resources as a behaviour that can perhaps be shaped by user-centered interventions. Perhaps some tools or design ideas may be used to help minimize or delay such behavior. We will propose three categories of different tools that may help students tackle this behavior, saving time and helping them to succeed in learning to program.

6.1.1 An IDE built to minimize student's time on external resources

To tackle our finding that students spend a large amount of time looking for external resources, future work could explore tools that allow learners to minimize wasted time; helping students spend that time better, focusing more on the process of writing code and reflecting rather than looking for more resources. For example, creating a special integrated development environment (IDE), it is possible to track the time that students spend coding. This IDE could track the time this app is on the foreground, log the keyboard strokes cadence (strokes per minute), and log the number of running attempts students make. Students may request some time off from the IDE by pressing a pause button. Students can also read the official Runestone textbook through an integrated browser window within the IDE. By keeping track of the time, we aim to make students spend more time coding, reflecting and learning. By keeping track of the keyboard strokes, we aim to make students actually write a code on their own without copying and pasting code from the web. By keeping track of the number of attempts to run their code, we aim to make students more aware of their effort and achievements. All of these variables are calculated and reported in a score within the IDE. When a specific milestone is reached, this IDE gives students some *free time* to look into external resources, but when time is up, the program reminds the user to return to the IDE and keep working on the assignment. If the student does not return, they would start losing points. If the IDE is running in the foreground, no points would be taken.

The student's goal is not only to solve the assignment using programming code, but also to obtain a higher score within the IDE. This score is computed by this IDE, and saved along with the student's solution to the assignment in a file that will be handed in to the teacher. The proposed IDE aims to promote students' exploration of different ideas and reward students that push hard, trying to solve the assignment on their own making mistakes, reflecting and correcting them; thus reducing weight on the *correct* solution and focusing more on the process of solving an activity that requires perseverance, trial and error, and reflection on their own errors.

6.1.2 An IDE that support students to get unstuck offering ideas

From our study, we observed that the majority of students used external resources to *get ideas, get inspiration or get unstuck* (44.1% in survey2, Table 4.1). Thus, future work might explore tools that allow learners to obtain these ideas or inspiration directly from the tool.

Having such available information could minimize students' time spent looking for external resources. This proposal consists of designing an integrated development environment (IDE) especially created to support programming assignments. In this IDE, the student loads a file, previously handed out by the instructor, with all the details of the students' programming exercise, including the problem to solve, requirements to meet (eg. code structures the student needs to use), the example output (a correct output for testing purposes), some hints (suggestions for students when they get stuck), and stages (different levels to measure students' effort and progress).

The idea of having this IDE is to provide better support for programming assignments than any generic-purpose IDE. For example, when students run their code, the IDE may automatically compare the code output with the instructor's correct output, informing students if the output does not coincide and in which part it did not. If students are stuck with no idea about what to do, they can request some hints just by pressing a "get hint" button. These hints would have been included by the instructor in the assignment file. These hints would be designed to apply scaffolded learning, providing general tips at first and more detailed hints later. These hints could also provide links to specific resources, such as the official textbook or curated videos when required. All the activity and the number of accessed hints would be logged. When the student moves forward, the IDE shows the current stage, starting with zero (initial), then moving on to one, two or three (assignment almost finalized). The stages would serve to indicate a perceived student's progress and effort. For example, writing 20 code lines and pressing *code run* button five times can lead students to move from stage one to stage two, thus accessing a new level of hints. Obtaining correct output for positive numbers, but not for negative ones, could lead to move to the next stage as well. The meta information logged by this IDE could also help teachers to know how difficult the programming assignment was for their students. With this information, teachers might craft better hints, tips and other useful details that could help students better, and with this help, make students require fewer external resources.

6.1.3 A browser plugin to minimize the use of external resources

Another proposed direction for future work is directly trying to minimize the time students spend on external resources. Using ideas from persuasive system design [106], we propose a browser plugin that limits the attention time students spend on external resources, highlighting that that time is a limited resource that should not be wasted. The idea is that this browser plugin would measure the number of queries someone does on google or the number of new links this person clicks on the Google result page. With this, it might detect that a student is getting into a *high reliance on google* mode by comparing how similar each google query is to one another.

Other helpful variables worth considering are: number of tabs opened at the same time, the speed at which new tabs are opened, the number of tabs opened without being visited,

or how quick a user scrolls down web pages, not reading them carefully. Through logging this data, this plugin may create a personalized dashboard that will display information to a specific student, comparing this information to that of previous weeks. This tool may help students reflect about the perils of heavily relying on external resources, suggesting that such behavior is a waste of time and not an optimal way to learn.

6.2 Future work: Helping learners from a CS Education pedagogical approach

The area of Computing Science includes investigating a wide variety of topics. Investigations in CS education research include how students learn about programming and problem solving, how novice programmers develop programming skills, how teachers address different teaching methods when teaching CS, how computer based tools can support learning, and about studying social issues connected to the usage of technology and computers, among many others [45] [36]. As we can see from this description, there is some overlap with the HCI field since both fields concur on helping novice users and learners to succeed. From the CSEd approach, we consider some pedagogical interventions that may help students overcome their high reliance on external resources.

6.2.1 Programming assignment design proposal through planned reflections when coding

One idea to minimize students' reliance on external resources is using planned reflection when coding. This would build up on the idea that the best way to get a good solution is to explore several possible solutions. Doing that requires time to reflect and test these possible solutions in order to see which one is better. Our proposal is that instructors design programming assignments that would be deployed into three parts: part one would ask students about general ideas regarding how to solve the assignment. Part two would ask students about the actual code for solving the assignment, and part three would ask students about lessons learned about the approach used to solve the task.

In part one, instructors would ask students about general ideas that they may use to solve the given assignment. These ideas would be brief and written in plain English, using high-level explanations or procedures, not programming code. The main questions in this part could be: *how do you plan to solve the assignment? What ideas do you think may work?*. Through this part, the instructors would promote students' reflection on high-level computational thinking and algorithmic strategy to solve a given problem. The high-level plain way to explain the ideas in English would make students comfortable enough to go as detailed or generic as they wish. Furthermore, there would not be any right or wrong ideas. The goal here is to make students start thinking about a problem before writing any code,

which is very important in order to build an strategy to solve any programming task. This part one would be required to be submitted at least three days before part two.

Part two would be about trying to solve the task using programming. The idea of this part is to request the programming code that solves the assignment, just like a regular assignment. The main difference with the regular assignment is that instructors would focus less on the *right* code and more on the exploration of ideas, trying to solve the exercises using different ways. So, there could be code that does not solve the problem completely from the computer perspective. Nonetheless, if instructors can see that students had an idea or approach that may be valid, instructors could give them some partial points.

Part three is about self-reflection, which is part of the Self-Regulation of Learning [86]. In this part, instructors would ask students to write about which ideas were OK, which ideas were wrong and what they learned from this assignment. Using this planned reflection approach, students may be less worried about the *"right code/solution"* (which they may be temped to get from the web), and more willing to explore their own solutions; making students think more about the problem before writing any code. By requesting to submit part one at least three days before the actual coding part, instructors would make students think about the task, giving them plenty of time to reflect and start coding earlier than with the regular assignment approach. An advantage of using this approach is that we can make students more interested in learning how their own code actually works, instead of just using someone else's code that students can grab from the web but may be hard to understand, although it may solve the assignment as required.

6.2.2 Accepting that students' reliance on external resources could be hard to minimize

Future research may have to accept the premise that we may not entirely change students' reliance on external resources given that Google and other types of search engines are intertwined with our daily lives [100] [102]. Thus, in our last proposal, we want to explore the idea that we might need to embrace the approach of using non-traditional web resources, such as *StackOverflow*, as official course resources, although they were designed for on-the-job troubleshooting and not for formal learning. We want to imagine an introductory programming course in which *StackOverflow* can be integrated as a core tool with guidance on how to use it properly. Indeed, there is an interesting prior work, conducted by Lopez-Nores et al [63] in Spain, in which the authors used *StackOverflow* as a core tool in an intermediate-level programming course. The authors sought to promote Q&A websites for programmers as the main source of reference for students, and turn the teacher into a permanent observer who delivers pertinent advice. The study authors found interesting results, such as advantages in terms of understanding of programming, problem-solving capabilities, and even greater clarity in communication and documentation, along with positive perceptions by students.

Our course design proposal is to build a course based on the *learning by doing constructivism* [37] and Case-Based Learning [7] [93] approach, trying to address the problems identified by Mitchel Resnick and Natalie Rusk [83] about the way students learn how to code nowadays. The focus shall be set on the guidance of their learning process rather than on transmission of contents. We propose building an introductory programming course using the following ideas:

1. Teaching introductory programming using Case-Based Learning approach. Case-based learning (CBL) is an established approach across disciplines where students apply their knowledge to real-world scenarios, promoting higher levels of cognition under Bloom's Taxonomy [55] [94]. CBL approach has been popular in business schools due to its hands-on nature. In CBL approach students typically work on case studies in groups, developing stories involving one or more characters or scenarios. Our proposal is letting students play and explore with real-world code they may find on web sites such as *GitHub* or *StackOverflow*, and learning to program through solving simple but meaningful tasks (for example, visualizing interesting data [6] or composing music [87]) under guidance of the course instructor.

2. Promote the exploration of ideas. Computers are not just simple tools, but mind amplifiers which develop and maximize people's potential for solving problems. Therefore, the exploration of ideas should be encouraged. For example, one activity could consist of tweaking code and predicting its output related to an idea. Another activity could be prototyping by mixing different codes (code mash-ups).

3. We suggest exploring the idea of changing the instructor's role, from that of an information-provider to something more similar to that of a sport coach, who encourages students to solve problems using computational creativity. Although this coach proposal has been explored before to teach how to code [66] [81], we believe that coaching suits better with the CBL approach while using real-world resources, such as Q&A resources. This approach of coaching, CBL and Q&A resources such as *StackOverflow*, is very similar to the way professional programmers work in the industry nowadays [26] [68] [97].

6.3 Limitations

Like any empirical study, ours has limitations. Although our data was gathered within a large institution, we should be cautious not to generalize our results too broadly to other scenarios.

We noted that the number of CS majors enrolled in the summer term was lower (9.5%) compared to the fall term (44.9%). It is to be noted that despite this variation, we did not observe different results reported by CS students, compared to non-CS ones. Another

limitation is that we used data from the first and second emergency-remote terms, when everything was uncertain and the reliance on external resources may have been unusually high. As the university teaching continued in emergency-remote settings for a while, students may have become more knowledgeable in that kind of teaching and learning modality. Although the vaccination campaigns worldwide have been ramp-up, and this emergency-remote settings may finish soon, we think that emergency-remote teaching may have some impact that may last on students and teachers in the near future. Another possible limitation is that we did not interview any teaching assistants or instructors. That may have helped us understand better some attitudes we observed among our study participants.

6.4 Future ideas for research

Here we want to present some ideas that may be worth further investigating. First, it would be interesting to see if our findings may be observed in other settings, such as in non-emergency fully online or hybrid programming courses. We also think that it would be interesting to make a comparison between students' behaviour when looking for external resources in emergency-remote programming courses and their behavior within in-person courses, in order to see if the high reliance on external resources that we found in this study also occurs in traditional on-campus university settings, in which asking peers, TA's or professors for help has lower social costs. Our research has an important qualitative component that helped us understand better some insights we observed along our surveys. Future work may be built upon the quantitative and experimental approach. Finally, we believe that it may be worth studying if there is any correlation between the level of reliance on external resources and the likelihood of late submissions [4], especially in online and emergency-remote courses that rely on students' self-monitoring of their own time and progress [110].

Chapter 7

Conclusions

We studied students' learning strategies when taking an emergency-remote introductory programming course. Our study supports that students' learning process is closely intertwined with external resources, regardless the quality and accessibility of the course provided resources. In particular, our study participants used Q&A forum resources more widely than online videos, as they were usually willing to find an exact answer to their problem instead of being willing to invest time enhancing their understanding. However, we noticed that our study participants reported they often wasted their time as they struggled to formulate relevant queries to locate relevant content, mainly due to their lack of expertise in the course subject. We analyzed some theories that may explain the high reliance on external resources and the behavior of wasting time to obtain an exact solution, as our study participants didn't seem to mind spending their time and energy sifting through as many resources as possible. A key concern raised from our findings is that students seem to neither reflect on their comprehension nor stop to ask for help, which may have repercussions on their overall learning. We suggested three tools from the HCI perspective, and discussed two educational approaches, along with some research venues that may be explored by the HCI and CS Education research communities to help students succeed in introductory programming courses, particularly in emergency-remote contexts.

Bibliography

[1] Rabe Abdalkareem, Emad Shihab, and Juergen Rilling. On code reuse from stackoverflow: An exploratory study on android apps. *Information and Software Technology*, 88:148–158, 2017.

[2] Saida Affouneh, Soheil Salha, and Zuheir N Khlaif. Designing quality e-learning environments for emergency remote teaching in coronavirus crisis. *Interdisciplinary Journal of Virtual Learning in Medical Sciences*, 11(2):135–137, 2020.

[3] Vincent Aleven, IDO Roll, Bruce M McLaren, and Kenneth R Koedinger. Automated, unobtrusive, action-by-action assessment of self-regulation during learning with an intelligent tutoring system. *Educational Psychologist*, 45(4):224–233, 2010.

[4] Dan Ariely and Klaus Wertenbroch. Procrastination, deadlines, and performance: Self-control by precommitment. *Psychological science*, 13(3):219–224, 2002.

[5] Roger Azevedo and Reza Feyzi-Behnagh. Dysregulated learning with advanced learning technologies. In *2010 AAAI Fall Symposium Series*, 2010.

[6] Lynne P Baldwin and Jasna Kuljis. Learning programming using program visualization techniques. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 8–pp. IEEE, 2001.

[7] Louis B Barnes, Louis B Barnes, Carl Roland Christensen, Louis B Barnes, Abby J Hansen, and Tony L Hansen. *Teaching and the case method: Text, cases, and readings*. Harvard Business Press, 1994.

[8] Anton Barua, Stephen W Thomas, and Ahmed E Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19(3):619–654, 2014.

[9] Debarati Basu, Sarah Heckman, and Mary Lou Maher. Online vs face-to-face web-development course: Course strategies, learning, and engagement. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1191–1197, 2021.

[10] Tamara Butler Battaglino, Matt Haldeman, and Eleanor Laurans. The costs of online learning. *Education reform for the digital era*, 1:1–13, 2012.

[11] Meltem Huri Baturay. An overview of the world of moocs. *Procedia-Social and Behavioral Sciences*, 174:427–433, 2015.

[12] Andrew Begel and Beth Simon. Struggles of new college graduates in their first software development job. *SIGCSE Bull.*, 40(1):226–230, March 2008.

[13] Jens Bennedsen and Michael E Caspersen. Failure rates in introductory programming: 12 years later. *ACM Inroads*, 10(2):30–36, 2019.

[14] Anne Bialowas and Sarah Steimel. Less is more: Use of video to address the problem of teacher immediacy and presence in online courses. *International Journal of Teaching and Learning in Higher Education*, 31(2):354–364, 2019.

[15] Alan F Blackwell. First steps in programming: A rationale for attention investment models. In *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, pages 2–10. IEEE, 2002.

[16] Joel Brandt, Philip J Guo, Joel Lewenstein, Mira Dontcheva, and Scott R Klemmer. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1589–1598, 2009.

[17] Joel Brandt, Philip J Guo, Joel Lewenstein, Mira Dontcheva, and Scott R Klemmer. Writing code to prototype, ideate, and discover. *IEEE software*, 26(5):18–24, 2009.

[18] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. Studying learning in the worldwide classroom research into edx's first mooc. *Research & Practice in Assessment*, 8:13–25, 2013.

[19] Matthew Butler, Michael Morgan, et al. Learning challenges faced by novice programming students studying high level and low feedback concepts. *Proceedings ascilite Singapore*, 1(99-107), 2007.

[20] Ruth Butler. Determinants of help seeking: Relations between perceived reasons for classroom help-avoidance and help-seeking behaviors in an experimental context. *Journal of Educational Psychology*, 90(4):630, 1998.

[21] Nicole Buzzetto-More. Student attitudes towards the integration of youtube in online, hybrid, and web-assisted courses: An examination of the impact of course modality on perception. *Journal of Online Learning and Teaching*, 11(1):55, 2015.

[22] John M Carroll and Mary Beth Rosson. Paradox of the active user. In *Interfacing thought: Cognitive aspects of human-computer interaction*, pages 80–111. 1987.

[23] Yunfan Chen and Ming Zhang. Mooc student dropout: Pattern and prevention. In *Proceedings of the ACM Turing 50th Celebration Conference - China*, ACM TUR-C '17, New York, NY, USA, 2017. Association for Computing Machinery.

[24] Tom Crick, Cathryn Knight, Richard Watermeyer, and Janet Goodall. An overview of the impact of covid-19 and "emergency remote teaching" on international cs education practitioners. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, page 1288, New York, NY, USA, 2021. Association for Computing Machinery.

[25] Georgios Digkas, Nikolaos Nikolaidis, Apostolos Ampatzoglou, and Alexander Chatzigeorgiou. Reusing code from stackoverflow: the effect on technical debt. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 87–91. IEEE, 2019.

[26] Ahmad Diyanati, Behrooz Shahi Sheykhahmadloo, Seyed Mostafa Fakhrahmad, Mohammad Hadi Sadredini, and Mohammad Hassan Diyanati. A proposed approach to determining expertise level of stackoverflow programmers based on mining of user comments. *Journal of Computer Languages*, 61:101000, 2020.

[27] Pierpaolo Dondio and Suha Shaheen. Is stackoverflow an effective complement to gaining practical knowledge compared to traditional computer science learning? In *Proceedings of the 2019 11th International Conference on Education Technology and Computers*, pages 132–138, 2019.

[28] Pierpaolo Dondio and Suha Shaheen. Is stackoverflow an effective complement to gaining practical knowledge compared to traditional computer science learning? In *Proceedings of the 2019 11th International Conference on Education Technology and Computers*, ICETC 2019, page 132–138, New York, NY, USA, 2019. Association for Computing Machinery.

[29] Brian Dorn, Adam Stankiewicz, and Chris Roggi. Lost while searching: Difficulties in information seeking among end-user programmers. *Proceedings of the American Society for Information Science and Technology*, 50(1):1–10, 2013.

[30] John Dutton, Marilyn Dutton, and Jo Perry. Do online students perform as well as lecture students? *Journal of Engineering education*, 90(1):131–136, 2001.

[31] John Dutton, Marilyn Dutton, and Jo Perry. Do online students perform as well as lecture students? *Journal of engineering education (Washington, D.C.)*, 90(1):131–136, 2001.

[32] Rylan Egan, Diana Cukierman, and Donna McGee Thompson. The academic enhancement program in introductory cs: a workshop framework description and evaluation. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pages 278–282, 2011.

[33] Ghada Refaat El Said. Understanding how learners use massive open online courses and why they drop out: Thematic analysis of an interview study in a developing country. *Journal of Educational Computing Research*, 55(5):724–752, 2017.

[34] Katrina Falkner, Rebecca Vivian, and Nickolas J.G. Falkner. Identifying computer science self-regulated learning strategies. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, page 291–296, New York, NY, USA, 2014. Association for Computing Machinery.

[35] Rebecca Ferguson and Doug Clow. Consistent commitment: Patterns of engagement across time in massive open online courses (moocs). *Journal of Learning Analytics*, 2(3):55–80, 2015.

[36] Sally Fincher and Marian Petre. *Computer science education research*. CRC Press, 2004.

[37] Catherine Twomey Fosnot. *Constructivism: Theory, perspectives, and practice*. Teachers College Press, 2013.

[38] Stuart Garner. Learning resources and tools to aid novices learn programming. In *Informing science & information technology education joint conference (INSITE)*, pages 213–222, 2003.

[39] Laura A Gelles, Susan M Lord, Gordon D Hoople, Diana A Chen, and Joel Alejandro Mejia. Compassionate flexibility and self-discipline: Student adaptation to emergency remote teaching in an integrated engineering energy course during covid-19. *Education Sciences*, 10(11):304, 2020.

[40] Michael H Goldhaber. The attention economy and the net. *First Monday*, 1997.

[41] Philip J Guo, Juho Kim, and Rob Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50, 2014.

[42] Allyson Fiona Hadwin and Philip H Winne. Study strategies have meager support: A review with recommendations for implementation. *The Journal of Higher Education*, 67(6):692–715, 1996.

[43] Anna Hansch, Katherine McConachie, Philipp Schmidt, Lisa Hillers, Christopher Newman, and Thomas Schildhauer. The role of video in online learning: Findings from the field and critical reflections. *Alexander von Humboldt Institut Fur Internet und Gesellschaft*, 2015.

[44] Taralynn Hartsell and Steve Chi-Yin Yuen. Video streaming in online learning. *AACE Journal*, 14(1):31–43, 2006.

[45] Orit Hazzan, Tami Lapidot, and Noa Ragonis. *Guide to teaching computer science*. Springer, 2014.

[46] Thomas T Hewett, Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. *ACM SIGCHI curricula for human-computer interaction*. ACM, 1992.

[47] Andrew D Hilton, Genevieve M Lipp, and Susan H Rodger. A technique for translation from problem to code. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pages 375–375, 2018.

[48] Charles Hodges, Stephanie Moore, Barb Lockee, Torrey Trust, Aaron Bond, et al. The difference between emergency remote teaching and online learning. *Educause review*, 27:1–12, 2020.

[49] Christopher D Hromalik and Tiffany A Koszalka. Self-regulation of the use of digital resources in an online language learning course improves learning outcomes. *Distance Education*, 39(4):528–547, 2018.

[50] Jane Hsieh, Michael Xieyang Liu, Brad A Myers, and Aniket Kittur. An exploratory study of web foraging to understand and support programming decisions. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 305–306. IEEE, 2018.

[51] Ting-Chia Hsu and Gwo-Jen Hwang. Effects of a structured resource-based web issue-quest approach on students' learning performances in computer programming courses. *Journal of Educational Technology & Society*, 20(3):82–94, 2017.

[52] Stuart A Karabenick. Perceived achievement goal structure and college student help seeking. *Journal of educational psychology*, 96(3):569, 2004.

[53] Zuheir N Khlaif, Soheil Salha, and Bochra Kouraichi. Emergency remote learning during covid-19 crisis: Students' engagement. *Education and Information Technologies*, pages 1–23, 2021.

[54] Rob Koper. How do students want to learn in online distance education? profiling student preferences. *International review of research in open and distributed learning*, 16(1):307–329, 2015.

[55] David R Krathwohl. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218, 2002.

[56] Sandeep Kaur Kuttal, Se Yeon Kim, Carlos Martos, and Alexandra Bejarano. How end-user programmers forage in online repositories? an information foraging perspective. *Journal of Computer Languages*, 62:101010, 2021.

[57] Samson Lausa, Rex Bringula, Annaliza E Catacutan-Bangit, and Cereneo Santiago. Information-seeking behavior of computing students while programming: Educational learning materials usage, satisfaction of use, and inconveniences. In *2021 IEEE Global Engineering Education Conference (EDUCON)*, pages 761–765. IEEE, 2021.

[58] Corinne Laverty, Brenda Reed, and Elizabeth Lee. The "i'm feeling lucky syndrome": Teacher-candidates' knowledge of web searching strategies. *Partnership: The Canadian Journal of Library and Information Practice and Research*, 3(1), 2008.

[59] Darren J Lipomi. Video for active and remote learning. *Trends in Chemistry*, 2(6):483–485, 2020.

[60] Allison Littlejohn. *Reusing online resources: a sustainable approach to e-learning*. Psychology Press, 2003.

[61] Dastyni Loksa and Andrew J Ko. The role of self-regulation in programming problem solving process and success. In *Proceedings of the 2016 ACM conference on international computing education research*, pages 83–91, 2016.

[62] Dastyni Loksa and Andrew J. Ko. The role of self-regulation in programming problem solving process and success. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, ICER '16, page 83–91, New York, NY, USA, 2016. Association for Computing Machinery.

[63] Martín López-Nores, Yolanda Blanco-Fernández, Jack F Bravo-Torres, José J Pazos-Arias, Alberto Gil-Solla, and Manuel Ramos-Cabrer. Experiences from placing stack overflow at the core of an intermediate programming course. *Computer Applications in Engineering Education*, 27(3):698–707, 2019.

[64] Zachary MacHardy and Zachary A Pardos. Evaluating the relevance of educational videos using bkt and big data. *International Educational Data Mining Society*, 2015.

[65] Gianclaudio Malgieri and Bart Custers. Pricing privacy–the right to know the value of your personal data. *Computer Law & Security Review*, 34(2):289–303, 2018.

[66] Pablo Martín-Ramos, Maria João Lopes, M Margarida Lima da Silva, Pedro EB Gomes, Pedro S Pereira da Silva, José PP Domingues, and Manuela Ramos Silva. First exposure to arduino through peer-coaching: Impact on students' attitudes towards programming. *Computers in Human Behavior*, 76:51–58, 2017.

[67] Bradley N. Miller and David L. Ranum. Beyond pdf and epub: Toward an interactive textbook. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '12, page 150–155, New York, NY, USA, 2012. Association for Computing Machinery.

[68] Seyed Mehdi Nasehi, Jonathan Sillito, Frank Maurer, and Chris Burns. What makes a good code example?: A study of programming q&a in stackoverflow. In *2012 28th IEEE International Conference on Software Maintenance (ICSM)*, pages 25–34. IEEE, 2012.

[69] Donald A Norman. *The psychology of everyday things.* Basic books, 1988.

[70] Donald A Norman and Stephen W Draper. User centered system design: New perspectives on human-computer interaction. 1986.

[71] Tanty Oktavia, Harjanto Prabowo, Suhono Harso Supangkat, et al. The comparison of mooc (massive open online course) platforms of edx and coursera (study case: Student of programming courses). In *2018 International Conference on Information Management and Technology (ICIMTech)*, pages 339–344. IEEE, 2018.

[72] Maria Papoutsoglou, Nikolaos Mittas, and Lefteris Angelis. Mining people analytics from stackoverflow job advertisements. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 108–115. IEEE, 2017.

[73] Steven Pinker. *The sense of style: The thinking person's guide to writing in the 21st century.* Penguin Books, 2015.

[74] David J Piorkowski, Scott D Fleming, Irwin Kwan, Margaret M Burnett, Christopher Scaffidi, Rachel KE Bellamy, and Joshua Jordahl. The whats and hows of programmers' foraging diets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3063–3072, 2013.

[75] Peter Pirolli and Stuart Card. Information foraging. *Psychological review*, 106(4):643, 1999.

[76] Peter LT Pirolli. *Information foraging theory: Adaptive interaction with information.* Oxford University Press, 2009.

[77] Luca Ponzanelli, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, and Michele Lanza. Mining stackoverflow to turn the ide into a self-confident programming prompter. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 102–111, 2014.

[78] T Portante and R Tarro. Paying attention. *Wired*, 5(9):114–116, 1997.

[79] Thomas W Price, Zhongxiu Liu, Veronica Cateté, and Tiffany Barnes. Factors influencing students' help-seeking behavior while programming with human and computer tutors. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 127–135, 2017.

[80] Tom Prickett, Julie Walters, Longzhi Yang, Morgan Harvey, and Tom Crick. Resilience and effective learning in first-year undergraduate computer science. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, pages 19–25, 2020.

[81] Naseem Rahman, David Nandigam, and Sreenivas Sremath Tirumala. Teaching computer programming with a coaching mindset. 2015.

[82] V Smrithi Rekha and S Venkatapathy. Understanding the usage of online forums as learning platforms. *Procedia Computer Science*, 46:499–506, 2015.

[83] Mitchel Resnick and Natalie Rusk. Coding at a crossroads. *Communications of the ACM*, 63(11):120–127, 2020.

[84] Allison M Ryan and Paul R Pintrich. " should i ask for help?" the role of motivation and attitudes in adolescents' help seeking in math class. *Journal of educational psychology*, 89(2):329, 1997.

[85] Allison M Ryan, Paul R Pintrich, and Carol Midgley. Avoiding seeking help in the classroom: Who and why? *Educational Psychology Review*, 13(2):93–114, 2001.

[86] Dale H Schunk and Barry J Zimmerman. *Self-regulation of learning and performance: Issues and educational applications.* Lawrence Erlbaum Associates, Inc, 1994.

[87] Ben Shapiro. Emerging paradigms for cs education and their implications for visual languages (invited keynote). In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–1. IEEE Computer Society, 2017.

[88] Adam C Short. Adaptive ides to support information foraging for novice programmers. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 235–236. IEEE, 2019.

[89] Derek Sleeman. The challenges of teaching computer programming. *Communications of the ACM*, 29(9):840–841, 1986.

[90] Tal Soffer and Anat Cohen. Students' engagement characteristics predict success and completion of online courses. *Journal of computer assisted learning*, 35(3):378–389, 2019.

[91] Supraja Srivastava, Jenny Jacob, Aby S Charles, Priyanka Daniel, John K Mathew, Pauline Shanthi, Kiran Devamani, Gowri Mahasampath, and Suganthy Rabi. Emergency remote learning in anatomy during the covid-19 pandemic: A study evaluating academic factors contributing to anxiety among first year medical students. *medical journal armed forces india*, 77:S90–S98, 2021.

[92] Anselm Strauss and Juliet Corbin. *Basics of qualitative research*. Sage publications, 1990.

[93] Jun Tan, Xianping Guo, Weishi Zheng, et al. Case-based teaching using the laboratory animal system for learning c/c++ programming. *Computers & Education*, 77:39–49, 2014.

[94] Errol Thompson, Andrew Luxton-Reilly, Jacqueline L Whalley, Minjie Hu, and Phil Robbins. Bloom's taxonomy for cs assessment. In *Proceedings of the tenth conference on Australasian computing education-Volume 78*, pages 155–161, 2008.

[95] Steven D Tripp. The nurnberg funnel: Designing minimalist instruction for practical computer skill, 1990.

[96] Hans Van der Meij. Constraints on question asking in classrooms. *Journal of Educational Psychology*, 80(3):401, 1988.

[97] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. Stackoverflow and github: Associations between software development and crowdsourced knowledge. In *2013 International Conference on Social Computing*, pages 188–195. IEEE, 2013.

[98] Morteza Verdi, Ashkan Sami, Jafar Akhondali, Foutse Khomh, Gias Uddin, and Alireza Karami Motlagh. An empirical study of c++ vulnerabilities in crowd-sourced code examples. *IEEE Transactions on Software Engineering*, 2020.

[99] Giovanni Vincenti, Scott Hilberg, and James Braman. Student preferences for consuming supplemental instructional material in cs0/cs1/cs2 courses. In *Society for Information Technology & Teacher Education International Conference*, pages 558–566. Association for the Advancement of Computing in Education (AACE), 2015.

[100] Yifan Wang, Lingdan Wu, Liang Luo, Yifen Zhang, and Guangheng Dong. Short-term internet search using makes people rely on search engines when facing unknown issues. *PloS one*, 12(4):e0176325, 2017.

[101] Christopher Watson and Frederick W.B. Li. Failure rates in introductory programming revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, page 39–44, New York, NY, USA, 2014. Association for Computing Machinery.

[102] Daniel M Wegner and Adrian F Ward. How google is changing your brain. *Scientific American*, 309(6):58–61, 2013.

[103] Christian Werner and Boris Girnat. Towards self-motivated learning in computer science education: Results of a questionnaire study. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, pages 26–32, 2020.

[104] Pedro A Willging and Scott D Johnson. Factors that influence students' decision to dropout of online courses. *Journal of Asynchronous Learning Networks*, 13(3):115–127, 2009.

[105] Robert J Williams, Rachel A Volberg, Rhys MG Stevens, Lauren A Williams, and Jennifer N Arthur. The definition, dimensionalization, and assessment of gambling participation. Technical report, Canadian Consortium for Gambling Research, 2017.

[106] Michael JA Wohl, Avi Parush, Hyoun Andrew S Kim, and Kristen Warren. Building it better: applying human–computer interaction and persuasive system design principles to a monetary limit tool improves responsible gambling. *Computers in Human Behavior*, 37:124–132, 2014.

[107] Tak-Lam Wong, Haoran Xie, Fu Lee Wang, Kai Tai Tang, Anthony Kong, and Reggie Kwan. Promoting self-regulated learning by online educational resources. In *2018 International Symposium on Educational Technology (ISET)*, pages 204–208. IEEE, 2018.

[108] Benjamin Xie, Erik Harpstead, Betsy DiSalvo, Petr Slovak, Ahmed Kharrufa, Michael J. Lee, Viktoria Pammer-Schindler, Amy Ogan, and Joseph Jay Williams. Learning, education, and hci. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, page 1–4, New York, NY, USA, 2019. Association for Computing Machinery.

[109] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: Measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, ADKDD '13, New York, NY, USA, 2013. Association for Computing Machinery.

[110] Meina Zhu, Curtis J Bonk, and Min Young Doo. Self-directed learning in moocs: Exploring the relationships among motivation, self-monitoring, and self-management. *Educational Technology Research and Development*, pages 1–21, 2020.

[111] Barry J Zimmerman. Attaining self-regulation: A social cognitive perspective. In *Handbook of self-regulation*, pages 13–39. Elsevier, 2000.

[112] IS Zinovieva, VO Artemchuk, Anna V Iatsyshyn, OO Popov, VO Kovach, Andrii V Iatsyshyn, YO Romanenko, and OV Radchenko. The use of online coding platforms as additional distance tools in programming education. In *Journal of Physics: Conference Series*, volume 1840, page 012029. IOP Publishing, 2021.

Appendix A

Survey1 questionnaire

In this chapter, we present the survey1 questionnaire used in this thesis. All questions were optional.

The symbol ( ) means that just one option can be selected at most. The symbol [ ] means one or several options can be selected.

1. **How old are you?**

   ( ) 18 - 19 years.
   ( ) 20 - 21 years old.
   ( ) 22 - 25 years old.
   ( ) old 26- 30 years old.
   ( ) 30 years old to up.
   ( ) (I prefer not to declare).

2. **What is your gender?**

   ( ) Female.
   ( ) Male.
   ( ) Non-binary/other.
   ( ) Prefer not to specify.

3. **What is your major?**

   (blank space is provided)

4. **What is your grade point average?**

   (blank space is provided)

5. **How much programming experience did you have prior to this course (if any)?**

   ( ) None.
   ( ) Less than 5 hours.

( ) More than 5, less than 30 hours.
( ) More tha 30 hours, less than 100 hours.
( ) More than 100 hours.

6. **What was the last time that you did any programming, before taking this programming course?**

   ( ) I never did any programming.
   ( ) Within the last two months.
   ( ) Within the last two semesters.
   ( ) Within the last 2 years.
   ( ) More than 2 years ago.

7. **What do you feel about your background in Computing Science and Programming?**

   ( ) I have a very strong background.
   ( ) I have medium or basic background.
   ( ) I had quite a good background but I have forgotten a lot.
   ( ) I feel comfortable using my computer (laptop / smartphone / iPad, etc.) but I have very weak (or almost none) background in programming.
   ( ) I am not too comfortable using a computer (laptop / smartphone, etc.) and I have very weak (or almost none) background in programming.

8. **How many ONLINE college/university-level courses have you taken prior to this course (not considering Spring 2020)?**

   ( ) 0.
   ( ) 1.
   ( ) 2.
   ( ) 3.
   ( ) More than 3.

9. **How many hours of study per week (outside of class time) do you spend on this programming course on average? (provide an estimate)**

   ( ) 1 hour or less per week.
   ( ) 2 or 3 hours per week.
   ( ) 4 to 5 hours per week.
   ( ) 6 to 7 hours per week.
   ( ) 8 hours or more per week.

10. **When I don't understand something in this course, I prefer to:** (drag or select 1 for your first choice, 2 for the second and 3 for the third choice):

    ( ) Review my notes or check the given slides.
    ( ) Use books (printed or e-books).
    ( ) Seek help from teaching assistant.
    ( ) Use the interactive textbook provided in this course.

( ) Watch youtube videos.
( ) Read relevant web pages in general.
( ) Post question into a forum with my classmates.
( ) Post question into a forum with people outside the course.
( ) Talk to the course instructor.

11. **On average, how many hours per week do you use interactive textbook provided in this course?**

    ( ) Less than one hour.
    ( ) 1-2 hours.
    ( ) 3-4 hours.
    ( ) >5 hours.
    ( ) (I am not sure).

12. (Conditional question: if the user marked less than one hour in the previous question)

    Please tell us the main reasons for not using the interactive textbook more often (blank space to be filled)

13. **If there is something that you value the most from the interactive textbook, what would it be?**

    [ ] The provided quiz that help me realize if I am learning or not.
    [ ] The information provided is very useful and complete.
    [ ] The fact that I can access this book anytime and anywhere.
    [ ] The fact that I can play with python code and program inside the textbook.
    [ ] Other (please specify) (blank space is provided).

14. **When are you most likely to use instructor suggested resources when studying for this programming course (if at all)?**

    [ ] Prior to the lecture.
    [ ] Right after when I do not understand the lecture.
    [ ] When I try to solve the course assignment and I can not.
    [ ] When I plan to study for a quizz or exam.
    [ ] (I don't use any web-based resources at all).
    [ ] Other (please specify)(blank space is provided).

15. **When are you more likely to search for external resources, if at all? (resources not suggested by the course instructor)**

    [ ] When the suggested resources are to broad/generic (not specific to my needs).
    [ ] When I want to see a different explanation or more details.
    [ ] When I want to see more examples or exercises.
    [ ] When I want to see more step-by-step instructions.
    [ ] When I just want to refresh my knowledge before an exam/quiz.
    [ ] (I prefer not to use external resources).

[ ] Other (please specify) (blank space is provided).

16. **Do you have any of these difficulties? / For what types of topics have you consulted external resources in this class so far?**

[ ] Difficulty in learning the python code syntax (commands, arguments, etc).
[ ] Difficulty in understanding programming terms and concepts.
[ ] Difficulty in learning the programming environment (repl.it, IDLE, etc).
[ ] Difficulty in finding errors in my code and fix them.
[ ] (I don't have any of these difficulties).
[ ] Other (please specify) (blank space is provided).

17. **So far, these are the most frequent difficulties I have experienced when solving a coding problem:**

[ ] I don't know where or how to start.
[ ] I get python errors when executing and I don't know why.
[ ] I don't know how to interpret the given problem to solve or what is expected to me to do.
[ ] I don't know how to test my code, I can't see if I did it wrong or right.
[ ] I don't understand the technical jargon associated with programming.
[ ] I don't know how to detail my problem to google, in order to get relevant answers.
[ ] (I don't experience any of these problems).
[ ] Other (please specify) (blank space is provided).

18. **What resources external to the course do you find most useful?**

[ ] Videos.
[ ] Forums, Q&A websites.
[ ] Blogs.
[ ] Interactive tutorials.
[ ] Ebooks.
[ ] Printed books.
[ ] Web sites.
[ ] Chats.
[ ] Other (please specify) (blank space is provided).

19. **I feel that I understand programming concepts better:**

( ) When I am able to solve the exercises/quizzes in a timely manner.
( ) When I think I am capable to explain to a friend.
( ) When I feel I can use the new information as a tool to solve new problems.
( ) When I am able to clarify conflicting information that created doubts in my head.
( ) Other (please specify) (blank space is provided).

20. **What are the most important aspects that a resource (a web page, YouTube video, etc) has to have in order to be considered useful for your learning**

process? (drag or select 1 for your first choice, 2 for the second and 3 for the third choice)

( ) It should clearly state the topic, the goal and the scope of the lesson.
( ) It should have solved exercises, step by step.
( ) It should have clear explanations, assuming that I am not familiar with the material or the previous information.
( ) It should cover the entire aspect of the topic, for example, every case or course of action.
( ) It should be specific and time restricted. That means it should cut to the chase of what I require to learn.
( ) It should have come from a well-known or high-quality source (speaker, publisher, top web site, etc).

21. **It is easy to me to find help for this course by using resources outside of the course materials shared in this programming course:**

( ) Always.
( ) Usually.
( ) Sometimes.
( ) Rarely.
( ) Never.

22. **How often do you use external sites that provide specific course resources? (Coursehero, Chegg, Slader, etc)**

( ) Always.
( ) Usually.
( ) Sometimes.
( ) Rarely.
( ) Never.

23. **What do you do after you find something useful for your course, if anything? (e.g., use it in my assignment, take notes, save a link, etc.)**

(blank space is provided).

24. W**hat are some challenges/difficulties you face in your study for this course?**

(blank space is provided).

25. **How does your experience in this course so far compare to in-person courses? What is easier or more difficult?**

(blank space is provided).

Appendix B

Interview demographics survey

In this chapter, we present the demographics survey questionnaire used to gather data from our interviewees. This survey was taken before conducting the interview.

The symbol ( ) means that just one option can be selected at most.

1. **How old are you?**

   ( ) 18-19 years old.
   ( ) 20-21 years old.
   ( ) 22-25 years old.
   ( ) 26-30 years old.
   ( ) 30 years old and up.
   ( ) (I prefer not to declare).

2. **When did you join this university?**

   ( ) Just enrolled.
   ( ) One year ago.
   ( ) 2 years ago.
   ( ) 3 years ago.
   ( ) 4 years ago.
   ( ) 5 years ago.
   ( ) More than 5 years ago.

3. **What is your gender?**

   ( ) Female.
   ( ) Male.
   ( ) Non-binary/Other.
   ( ) (I prefer not to disclose).

4. **How much programming experience did you have prior to this course (if any)?**

( ) None.
( ) Less than 5 hours.
( ) More than 5, less than 30 hours.
( ) More than 30, less than 100 hours.
( ) More than 100 hours.

5. **What is your current (or projected) major?**
   (blank space is provided)

Appendix C

Interview questionnaire

In this chapter, we present the semi-structured interview questionnaire used in this thesis. All questions were optional.

Background/ Warm-up Questions:

1. Tell me a little about yourself - what's your major, what year are you in?
2. What motivated you to take this programming course?
3. Tell me a little about your experience with the course so far.
4. When learning this programming course, you have several resource options: your own course notes; resources created by the teacher (slides, for example), resources suggested by the teacher (interactive textbook) and external resources, not suggested by the teacher. Can you tell us how you start studying for solving a new programming exercise? What resource do you use to start? What do you do next?
5. Our survey showed that students use a lot of external resources (mostly Q&A, general websites, and video), in addition to those suggested by the course instructor. I am going to ask you a little bit about your own experience in using these resources. What strategies work for you to search new resources and why?
6. How do you judge that the resource is relevant or not?
7. Which one is the most important feature you value the most and why?
8. What are some of the disadvantages of each resource type?
9. Several students take and review their notes for learning programming. Do you take notes? How do you use your notes when studying online and using different resources?
10. How often do you use the provided interactive textbook (if at all)? Please tell us your opinions about the class textbook.

11. Ok, now I want you to think about the way you realize when you are learning, and the process of reflection that helps you to make some adjustments when you don't. Think about when you are studying for your midterm, when your study session unfolds, How self-aware are you? For example, how do you know that you have studied enough for that day?

12. Think about when you are working on a programming assignment, how do you decide that you have solved the problem accurately?

13. Some students mention that they waste a lot of time searching for relevant videos and looking up Q&A sites for answers. When using internet resources for studying, how easy or hard is for you to realize how much time you are spending on those resources?

14. Can you think of a recent situation where you shared a resource with somebody or asked somebody to share resources they found with you? What triggered you to do that?

15. Let's suppose that you need to pay $10 to access any resource online, except resources provided by the instructor. How would it change your learning approach for this course?

16. One of the most frequent difficulties that students told us they face when learning to program is that often they don't know how to interpret the given problem to solve or they don't know what is expected for them to do. Have you experienced any of those problems?

17. Can you tell us how you control the speed of the information you are obtaining from external resources?

18. What is the most rewarding part of learning computer programming and why (if any)?

19. Is there anything else you would like to share with us about your experience in learning programming and using online resources?

Appendix D

Survey2 questionnaire

In this chapter, we present the survey2 questionnaire used in this thesis. All questions were optional.

The symbol ( ) means that just one option can be selected at most.

1. **How old are you?**

   ( ) 18 - 19 years old.
   ( ) 20 - 21 years old.
   ( ) 22 - 25 years old.
   ( ) 26- 30 years old.
   ( ) 30 years old to up.
   ( ) (I prefer not to declare).

2. **What is your gender?**

   ( ) Female.
   ( ) Male.
   ( ) Non-binary/other.
   ( ) Prefer not to specify.

3. **What is your major?**

   (blank space is provided)

4. **How long have you been a student at this university?**

   ( ) 1 year or less.
   ( ) 2 years.
   ( ) 3 years.
   ( ) 4 years.
   ( ) 5 years.
   ( ) more than 5 years.

5. **Which of the following best describes the reason you enrolled in this course?**

   ( ) It is a requirement for my major/minor.
   ( ) I think it will be valuable for my professional development.
   ( ) Someone suggested that I should take it / I am curious about programming.
   ( ) I am evaluating doing a major/minor in Computer Science.
   ( ) Other (please specify) (blank space is provided).

6. **To what extent do you see yourself writing code as part of your future job?**

   ( ) Extremely unlikely.
   ( ) Unlikely.
   ( ) Neutral.
   ( ) Likely.
   ( ) Extremely likely.

7. **How much programming experience did you have prior to this course (if any)?**

   ( ) None.
   ( ) Less than 5 hours.
   ( ) More than 5, less than 30 hours.
   ( ) More than 30, less than 100 hours.
   ( ) More than 100 hours.

8. **Beyond the course textbook and recommended resources provided in this course, to what extent do you feel the need to look up more programming resources (on your own)?**

   ( ) Never,
   ( ) Rarely,
   ( ) Sometimes,
   ( ) Often.
   ( ) Always.

9. **How frequently do you look up webpages (eg. pythonbasics.org) ON YOUR OWN to help yourself with this course?**

   ( ) Never.
   ( ) Rarely.
   ( ) Sometimes.
   ( ) Often.
   ( ) Always.

10. **To what extent would you say webpages that you look up ON YOUR OWN are actually helpful for your learning in this course?**

( ) Extremely helpful.
( ) Very helpful.
( ) Somewhat helpful.
( ) Not so helpful.
( ) Not at all helpful.

11. **When looking up webpages related to programming concepts, what do you value the most? (pick your most relevant reason)**

( ) The web page look and feel is well done.
( ) The web page comes up in first places on the google results.
( ) The web page has solved exercises, step by step.
( ) The web page content is easily to be search in order to get an answer to my question.
( ) The web page content is specific and time-constrained. I don't want to spend too much time reading.
( ) The web page content is comprehensive, including everything required to understand the lesson topic.
( ) The reliability of the source/author of the webpage is high.

12. **How frequently do you look up online videos ON YOUR OWN to help yourself with this course (eg. YouTube)?**

( ) Never.
( ) Rarely.
( ) Sometimes.
( ) Often.
( ) Always.

13. **To what extent would you say online videos that you look up ON YOUR OWN are actually helpful for your learning process in this course?**

( ) Extremely helpful.
( ) Very helpful.
( ) Somewhat helpful.
( ) Not so helpful.
( ) Not at all helpful.

14. **When looking for a new online video related to programming concepts, what do you value the most? (pick your most relevant reason)**

( ) The video has a clear explanation of the topic.
( ) The video is popular (e.g. high number of views or likes).
( ) The number of comments of the video is high.
( ) The number of solved or proposed exercises the video presents is high.
( ) The video length: shorter is better.
( ) The video has a clear description of the topic and goals.

( ) Good video production (video edition, resolution, audio and overall quality).
( ) The reliability of the source/author of the video is high.

15. **How frequently do you look up answers on forums or Q&A sites ON YOUR OWN to help yourself with this course (eg. StackOverflow)?**

( ) Never.
( ) Rarely.
( ) Sometimes.
( ) Often.
( ) Always.

16. **To what extent would you say forums and Q&A posts that you look up ON YOUR OWN are actually helpful for your learning in this course?**

( ) Extremely helpful.
( ) Very helpful.
( ) Somewhat helpful.
( ) Not so helpful.
( ) Not at all helpful.

17. **When looking for a answers on forums or Q&A sites related to programming concepts, what do you value the most? (pick your most relevant reason)**

( ) The question in that post is very similar to the question I have right now.
( ) The provided solution or included code in that post is easy to understand.
( ) The provided solution or code example in that post uses only elements that are part of my course topics (eg. it uses only while loops instead of other advanced programming methods).
( ) The provided solution is specific and time-constrained. I don't want to spend too much time reading.
( ) The provided solution to that post is upvoted high by a large number of users, helping to weed out bad answers or flawed code.
( ) The answer to that post includes code examples that I can try, reuse or adapt.
( ) The provided answer to that post gives me new ways to understand my problem or improve my quest for new resources.
( ) The reliability of the source/author of that Q&A answer is high.

18. **When I don't understand something in the course, I prefer to (pick your three most relevant choices)**

( ) Review my own course notes.
( ) Check the given slides or recorded lecture.
( ) Use the interactive textbook provided in this course.
( ) Use ebooks (eg. PDFs).
( ) Use printed books.

( ) Seek help from my instructor/TA,.
( ) Join to a peer tutor session.
( ) Watch videos (eg. YouTube).
( ) Use webpages in general (eg. pythonbasics.org).
( ) Post question into my course forum (eg. Canvas discussion forum).
( ) Use Q&A websites (eg. StackOverflow).

19. **When solving a programming assignment, what do you value the most? (pick your most relevant reason)**

( ) Solving the exercises in the shortest possible time .
( ) Solving the exercises using the lowest number of lines of code.
( ) Solving the exercises looking at the lowest number of web resources.
( ) Solving the exercises trying to make them run with the lowest number of attempts.
( ) Solving the exercises repurposing the lowest number of coding examples from the web.
( ) Solving the exercises without asking anyone.

20. **When studying for this course, how do you know if you have actually understood the relevant programming concepts? (pick your most relevant reason)**

( ) I feel comfortable with the new course material, it does not create confusion into my head.
( ) I am able to solve several exercises from others.
( ) I don't have to spend too much time studying.
( ) I can "play" with my code, tweaking, adding features and predicting their output.
( ) I don't require to access a large number of external resources to solve the exercise.
( ) I can explain it to my class mates.
( ) I am able to solve the problem in few lines or in few executing attempts.
( ) I am able to create my own programming exercises and solve them.

21. **To what extent would you agree with this statement: I feel like I waste a lot of time looking up resources rather than studying or working on my assignments**

( ) Strongly agree.
( ) Agree.
( ) Neither agree nor disagree.
( ) Disagree.
( ) Strongly disagree.

22. **How easy or hard is it for you to find the right external resources for your learning process in this course?**

( ) Very easy.
( ) Easy.

( ) Somewhat easy.
( ) Neither easy nor difficult.
( ) Somewhat difficult.
( ) Difficult.
( ) Very difficult.

23. **When are you most likely to abandon or give up a particular external resource? (pick your most relevant reason)**

    ( ) When there is a lack of code examples.
    ( ) When I run out of time.
    ( ) When I can't find the exact answer to my question.
    ( ) When the content is not engaging.
    ( ) When the provided code examples don't run properly.
    ( ) When it is too long to read/watch.
    ( ) When there is a lack of theory or explanation of the provided code or terminologies.

24. **In what cases do you rely on external resources available on the web? (beyond your own course notes or resources that the course instructor recommends). Please explain.**

    (blank space is provided)

25. **What strategies do you follow to look up new external resources (eg. webpages, videos, Q&A answers, etc) when you are studying or doing assignments for this course?**

    (blank space is provided)

26. **What are some challenges of using external resources that you look on your own (if any)?**

    (blank space is provided)

Appendix E

Coding scheme for open-ended questions

In this chapter, we present the open-coding scheme we used to process the open-ended questions presented in our findings.

Table E.1: Survey2: Codes and categories from students' answers to the open-ended question: *in which cases do you rely on external resources available on the web? (beyond your own course notes or resources that the course instructor recommends)*

C	**Meaning**	**N**	**Category**	**N**	**%**
g	When I am stuck (get hints, course of action or ideas).	10	When I want to get hints, explore different methods or find ideas for inspiration.	41	44.1%
f	Check different methods or ideas for achiving a solution.	13			
e	Check extra topics not covered in class (modules, concepts, etc)	16			
i	When no one can help me with my question.	2			
a	Incomplete explanation (theory) or basic previous material that they don't know.	24	When the perceived prior knowledge or course resources are deemed insuficient.	36	38.7%
j	When I want to get extra practice.	1			
b	Few examples provided by the course material or lack of variation on them	11			
c	Seek explanation for code errors, functions or programming syntax.	11	When Google is perceived as a handy book index.	16	17.2%
h	t is more convenient look into Google than sift over tons of materials.	5			

Table E.2: Survey2: Codes and categories from students' answers to the open-ended question: *what strategies do you follow to look up new external resources (eg. webpages, videos, Q&A answers, etc) when you are studying or doing assignments for this course?*

C	**Meaning**	**N**	**Category**	**N**	**%**
c	Search the exact coding error they got in the python IDE.	2	Using a search engine for clarification, solutions, examples or ideas.	52	62.7%
f	Having an idea of a "finished program" or program look like then start looking examples similar to it.	6			
g	Basic searching strategies using google. Eg. ' Python how to (something)", or "how to do (something) in python".	41			
j	Read the lectures and try code first, then go to google when specific requirements.	3			
a	They have a specific set of good or trusty external sites that they check preferently (stackoverflow, geek4feeks, etc).	21	Reduce the uncertaintly by relying only on specific internet sites (youtube excluded).	29	34.9%
m	Ask directly to my peers about good external resources.	1			
b	Check general videos basic concepts of the topic to better understand the code.	7	Use video (youtube) resources that are perceived to be easier to follow.	9	10.8%
i	Follow some specific people that explain well programming concepts on youtube	2			

Table E.3: Survey2: Codes and categories from students' answers to the open-ended question: *what are some challenges of using external resources that you look on your own (if any)?*

C	Meaning	N	Category	N	%
c	Hard to understand the resource or written code logic, not too much explanation or examples.	8	Challenges with resource comprehension (lack of clarity, beyond student's level, etc)	40	60.6%
d	More advanced functions that I was taught in the course, beyond my understanding.	22			
e	Difficult to determine whether the resource is relevant to me: Sometimes they are too in-depth.	10			
f	Lot of garbage you need to sift in order to get somethiing useful.	5	Challenges with the required effort/time to get good resources	26	39.4%
g	It requires too much time to get a resource that is useful for me.	3			
a	Hard to get good answers for the exercise I'm trying to solve or answers to my question.	9			
h	Lack of trust: The answer may not run when I am running it (outdated code, lack of libraries, different version, etc).	8			
m	It is hard to avoid plagiarism when using someone's else code.	1			
j	It is hard to convert my question into word to put into google search engine.	5	Challenges with terminology or formulating questions	7	10.6%
k	hard to make questions over the resource I am looking to.	2			
b	Understand terminology or jargon	7			