Utilization of Learning Resources by Undergraduate-level Students in Computer Programming Courses: An Exploratory Study

MAIRA KOTSOVOULOU
Information Technology Department
Deree - The American College of Greece
Gravias 6, Agia Paraskevi
GREECE
mkotsovoulou@acg.edu http://www.acg.edu/faculty/maira-kotsovoulou/

VASSILIA STEFANOU
Computer Information Systems Department
Deree - The American College of Greece
Gravias 6, Aghia Paraskevi - Athens
GREECE
vstefanou@acg.edu http://www.acg.edu/faculty/vassilia-stefanou/

DESPINA MAKRI
Information Technology Department
Deree - The American College of Greece
Gravias 6, Agia Paraskevi
GREECE
demakri@acg.edu http://www.acg.edu/faculty/despina-makri

Abstract : - This qualitative exploratory study, based on the theoretical principles of Grounded Theory, aims to explore students' perceptions of the utilization of learning resources in computer programming courses. The study employed an open class discussion, a survey, and semi-structured interviews for data collection. Data analysis was performed in an iterative and deductive manner, using open, selective and theoretical coding. Two coding schemes were developed from this process; one that conceptualizes the 'reason' and one that identifies the 'selection criteria'. Finally, the theoretical framework that emerged outlines relationships between the reasons why, and the purposes for which, students utilize learning resources.

Key-Words – learning resources, exploratory study, computer programming, grounded theory, educational resources

1 Introduction

The Internet has progressed from a simple networking scheme to a powerful and sophisticated communication mechanism that supports the dynamic exchange of information and ideas in a multimedia environment. As such, it has served as an ideal conduit to enhance and support traditional teaching and learning activities and provide innovative opportunities to optimize and further the learning process. Differences in learning styles may drive the preference of one learning resource over another, since most students are visual as opposed to verbal learners (Fowler, Allen, Armarego, & Mackenzie, 2000); however, due to the vast variety of learning resources available on the Internet, it is imperative that we take into account the students' reflections on their learning preferences.

The purpose of this exploratory study is to gain an insight into the types of resources students use to facilitate the learning process, as well as to develop a conceptual framework of the relationship between the types of learning resources students use when learning how to program and the selection criteria they apply in choosing them.

2 Grounded Theory Overview

Since the aim of this study is to understand the research situation, and given that there was no pre-formulated hypothesis, Grounded Theory seemed the most appropriate methodology in approaching the research question.

Grounded Theory, as defined by its founders Barney Glaser and Anselm Strauss (1967), is a methodology used to create a theory of a phenomenon grounded in systematically gathered and analyzed data. Compared to other research methods, Grounded Theory works in a cyclic, iterative fashion with frequent interplay between data collection, data coding and analysis, and theory development. The researcher may enter the field to discover the perception of participants (students) about learning resources and analyze the different ways that they approach their learning. "*The Grounded Theory process is both inductive and deductive. Inductive, in that instead of starting with a hypothesis or theory relevant, theoretical concepts are allowed to emerge from the data during the coding and categorization process.*" (Selvaraj & Fields, 2009, p. 472)

In a qualitative study, different sampling techniques can be used to involve participants in the study: selective, convenience, purposeful, theoretical, volunteer, convenient, nominated etc. Morse (1991) suggests that four types of sampling are most appropriate: the purposeful (theoretical) sample, the nominated sample, the volunteer sample and the sample that consists of the total population. The researcher has a research question in mind, prior to the initiation of the research project, and thus also possesses a specification of the kinds of subjects required by the study (Coyne, 1997). In this exploratory study, the subjects should fit the profile of undergraduate level students attending a programming course in Java. The decision for the sampling was made prior to the initiation of this study because of time, space and identity constraints. According to Sandelowski et al. (1992), this is selective sampling, while "*theoretical sampling refers to a sampling decision made on analytic grounds developed in the course of a study" (p. 302).*

Literature review shows that, although the research might begin with selective sampling from the population, theoretical sampling can be used to limit and further identify participants, according to the needs of the study.

Data collection techniques that are considered mainly qualitative include, but are not limited to: interviews, observations, notes, and discussions. Through immediate analysis of the data, conceptual codes are created. Grounded Theory is based on the conceptualization of data through coding. (Jones, Kriflik, & Zanko, 2005). With continuous data collection and by performing constant comparisons, codes are grouped to form meaningful categories. These categories aim to "*explain with the fewest possible concepts, and with the greatest possible scope, as much variation as possible in the behavior and problem under* study" (Glaser 1978)

Analysis occurs immediately and along with coding, and consequently goes through different levels of abstraction and deduction. Since the codes emerge from data analysis and the codes generate categories, an on-going sampling procedure should be performed. Theoretical sampling aims at directing the selection of new participants that will generate fresh data relevant to the work in progress. Glaser 1992 states that '*the general procedure of theoretical sampling is to elicit codes from the raw data from the start of data collection through constant comparative analysis as the data pour in. Then one uses the codes to direct further data collection, from which the codes are further developed theoretically with properties and theoretically coded connections with other categories until, each category is saturated. Theoretical sampling on any category ceases when it is saturated, elaborated and integrated into the emerging theory'* (p. 102). (Glaser, 1992)

Emerging categories are constantly refined until they are saturated and enable researchers to work towards the refinement of the theoretical framework that they were building during the process.

Is worth noting that continuous process of data collection → coding → analysis → theorizing is not directed by literature review. On the contrary, Strauss and Corbin and Glaser suggested that literature review should supplement the study and can be treated as another source of data to be integrated into the comparative analysis process.

One of the many considerations of this study was the appropriateness of using a quantitative method, such as a survey, to aid the collection of data for a methodology, which is mostly perceived as qualitative. An extensive literature review of the quantitative/qualitative debate shows that a researcher can use mixed methods as a means to attain meaningful and valid results and to answer his/her research questions. Any quantitative measure can be expressed qualitatively and any qualitative measure can be expressed in a quantitative manner. Robson (2003), Creswell (2003) and Krauss (2005) support that a mixed methodology will lead us to a comprehensive understanding of the problem and extract meaning from "the real world".

In accordance with the above, Glaser and Straus (1967, 1994) mention that using quantitative methods is not contradictory to the generation of Grounded Theory but can in fact "*lead to new* *strategies and styles of quantitative analysis.... that will bring out the richness of quantitative data that is seen only implicitly while the focus remains on verification"* (Glaser, 1994, p. 198)

Additionally, Strauss and Corbin (1998) maintain that the *"aim of theorizing is to develop useful theories. So, any methodology, whether qualitative or quantitative, is only a means for accomplishing that aim. We do not believe in the primacy of either mode of doing research."* Along the same lines, Yin (2003) stipulates that research should be assessed in terms of its context and the accuracy with which the methods have been used.

An exploratory study begins with a general subject area, which needs investigation. In line with the Grounded Theory, Glaser and Strauss (1967) support that, although the theoretical/conceptual framework should not be pre-conceived for a theory to emerge from the data, the researcher may begin by defining a few 'key principal features' of the situation under investigation.

3 The Research: General Design and Methodology

The process of learning how to program typically involves a combination of attending lectures, studying and practicing, in order to gain knowledge and acquire programming skills. In order to study and practice, students use a number of different resources, which are mainly found online. The main purpose of this study is to answer the following research questions: a) which learning research students use as an additional aid to study, understand and practice programming? And b) which are the selection criteria applied?

In order to answer the research questions, the following **'key principles'** needed to be investigated**:**

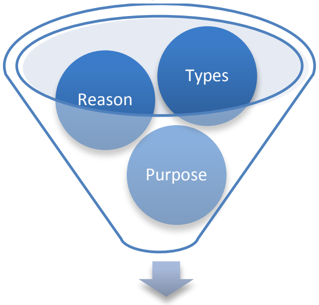
a) The types of learning resources used by students in my current programming classes ("what is used"),
b) The purpose students access additional resources ("how it is used")
c) The reasons students prefer to use one resource over another, that represent the selection criteria. ("why it is used")

At a successive stage, we will try to link the various emerging concepts together in order develop a conceptual framework. (Seibold, 2002)

3.1 Data collection

For data collection, three techniques were used: an open-class discussion, a survey and multiple semi-structured interviews. These were all performed in the English language, since participants are students of an English-speaking institution.

Fig. 1



Relationships?

3.1.1 Open class discussion

We initiated an open class conversation with our students, in which we posed the general research question. Students were encouraged to openly discuss their perceptions, while we kept notes of the main concepts. It was made clear to students that this discussion was not at all related to their course summative assessment and that their expressed opinions would not affect in anyway their course grade, which is calculated explicitly by a coursework assessment (40%) and a final exam (60%). After this preliminary investigation, we recorded and categorized all the data that were related to the phenomenon under investigation, and thus related to the key principle features defined at the beginning of this study. The theoretical framework evolved, and the types of learning resources identified and used by these students were added to the framework. Of course, further investigation was necessary to explore which of these resources were mostly used, which were perceived as learning tools, as well as to identify the applicability and context in which they were used.

3.1.2 Survey

The survey link was emailed to all 20 students attending an introductory java programming course and all 20 students attending an advanced java course. The students were briefed about the nature and scope of the survey and informed that their participation was not in any way obligatory. The main purpose of this survey was not to make generalizations to populations but rather to obtain an insight into the phenomenon under investigation by *"maximizing opportunities to discover variations among concepts and to densifying categories in terms of their properties and dimensions"* (Glaser, 1978, p. 124). This exploratory survey also aimed to identify students who were willing to be interviewed for further exploration of the case, these students emailed the researcher demonstrating their intention to participate. Since the survey was anonymous, it was assumed that students would feel more willing to answer questions regarding their study habits in an honest and truthful fashion, unbiased by concerns of overly exposing their thoughts to a tutor possibly viewed as a figure of authority.

3.1.3 Semi- structured Interviews

Eight individual, semi-structured interviews were performed with volunteers identified via the survey. The purpose of these interviews was to establish a more qualitative understanding of the type and breadth of resources used by students, the context in which they are used, and the reasons or perceived justifications as to their context-specific applicability. Three main categories of questions were directed at the participants: 1) questions regarding which resources they used (e.g. “There are many available resources and tools to aid students in learning programming. We are interested in which of these learning resources you have used during this course”), in order to trigger students’ responses and factually establish a basis for further, qualitative discussion, 2) questions exploring the cases or scenarios in which a particular resource was used and why the participant believed the resource was most appropriate or applicable for that case (i.e. “What do you reference when you want to study programming?” or “What do you do if you get stuck while you are programming?” or “How did this resource support you in your learning and why do you consider this resource as being more helpful than other alternatives?”), and 3) follow-up questions to clarify students’ statements to the above (i.e. “Can you please clarify that?” or “What do you mean when you say …?”).

3.5 Coding

Grounded Theory uses three levels of coding: initial (open-coding), selective and theoretical. The data collected from the open class discussion were coded using open coding. After the collection of the survey results, we moved on to the second level of coding. Selective coding allowed for filtering and determining the concepts most relevant to the research. The most frequently used resources were the ones that needed further exploration, in order to gain a deeper understanding of students’ perceptions, habits and preferences. Categories began to become denser, effectively becoming what Glaser defines as core categories. The first stage of interviews took place and theoretical coding provided criteria to assist in the development of conceptual relationships between categories and their relevance to the literature. Glaser (1978) indicated that theoretical sampling occurs when “the analyst jointly collects, codes, and analyzes his data and decides what data to collect next and where to find them, in order to develop his theory as it emerges” (p. 36). Following Glaser’s directions and as additional data emerged, a second phase of interviews took place.

3.1.4 Analysis

The open class discussion, which was the first phase of this research, aimed at determining an initial definition of the types of resources which are generally ‘known’ or used by the students to assist them in learning how to program. The discussion was initiated using the following assertion and question:

“There are multiple sources of information which you, as a student, can use while striving to learn programming. The Internet can provide most of them as easily and as quickly as possible, but traditional resources, such as your textbook, classmates and tutors, can help you as well. Have you used any learning resources in the duration of this course? Or are there any learning resources that you could have used?”

The students differentiated between studying and problem solving (troubleshooting) and they argued that they approach these two discrete purposes in a different manner.

The discussion produced the following list (in alphabetical order) of the Types of Resources:

- Blogs
- Books/e-books
- CBTS
- CMS (blackboard)
- Forums
- Hand-on exercises
- Lectures
- Peer tutoring/discussion (on-line/off-line)
- Programming examples (ready pieces of code)
- Tutors/Instructors
- Tutorials (videotaped lectures, screencasts)

And the following list of the Purposes:

- Process of Studying
- Process of Solving Programming Exercises/Troubleshooting

This list served as the basis for the on-line questionnaire that was to follow. Since the researchers are is also the tutors of the participants, we considered the survey as being more unbiased by student concerns due to the anonymity of the respondents. The survey produced the following findings: A total of 40 students majoring in Computer Information Systems and/or Information Technology, at the American College of Greece were invited to take part in the survey. In order to obtain results that could be deemed representative across varying degrees of participants' field-related academic proficiency, the 40 students were selected from courses spanning introductory and advanced levels – specifically, all 20 students enrolled in the entry-level Java programming class ("Introduction to Programming, Java I") and all 20 students enrolled in the advanced Java programming class ("Object Oriented Programming, Java II") were asked to participate in the survey. Within the one-week time frame allocated for survey completion, 15 students had completed the questionnaire provided to them via email, thus giving a final response rate of 37.5%. Furthermore, 8 students indicated their willingness via the survey to act as interview candidates and were thus identified as potential interview subjects.

Six (6) out of fifteen (15) students who participated in the survey were enrolled in the entry-level Introduction to Java programming class, while nine (9) were enrolled in the Objected Oriented Programming class.

In order to potentially qualify survey findings in the context of correlative effects, the questionnaire initially requested that participants specify their perception as to the general degree of difficulty they personally face in learning how to program in Java. Based on the survey responses, it was apparent that a relatively even distribution existed within the midrange levels of difficulty ("difficult" to "easy"), with minimum skew at the extremes ("very difficult" and "very easy/natural").

Most of the questions in the questionnaire required students to rate the resources on a 5-level scale, from "very useful" to "useless" with "neutral" being in the middle. The option "I have not tried it" was also included. What the students find useful or not so useful is based on how they perceive usefulness and according to Davis F. (1989) "perceived usefulness is the degree to which a person believes that using a particular system would enhance his or her job performance"

In response to the question of perceived usefulness of specific traditional resources when studying Java, students indicated a clear bias towards lab exercises/hands-on practice, with professor-led lectures themselves a close second. Almost all of respondents to this question gave the highest rating of "very useful". Class notes and group study were similarly positioned as next-best alternatives within the preference rankings of the respondents, while, interestingly enough, textbook examples/exercises and textbook theory were at the bottom of the ranking. In fact, textbook theory was actually rated from "neutral" to "useless" by 10 respondents.

When asked to rate specific traditional resources in terms of perceived usefulness in solving programming problems (i.e. Java coursework assignments), students demonstrated a distinct preference for lab exercises/hands-on practice, with 12 of the respondents regarding it as "very useful" and indicating that it constituted their first course of action. Class notes, consultations with instructors, and group study were ranked second, third, and fourth, respectively. Once again, textbook examples/exercises and textbook theory were near the bottom of the ranking, while Java manual (IDE Help) was the least preferred traditional resource.

When comparing rankings of the same traditional resources within the context of studying Java versus solving specific Java programming problems, the survey results illustrate that non-trivial perceptions of "uselessness" exist for certain resources (namely IDE Help and textbook theory) in the case of studying Java, whereas in the case of solving specific Java programming problems, these perceptions are limited and can be considered negligible.

In response to the question of perceived usefulness of specific e-resources when studying Java, students indicated a strong preference for tutorials, with the three related subcategories as the most preferred in the following order: video tutorials (screen-cams and how-to), online tutorials (text and images), and video tutorials (recorded theory lectures). Ratings of "very useful" were given by 12 of the respondents to video tutorials (screen-cams and how-to). Blackboard was rated as the next most popular e-resource, although its perceived strength was in the same proximity as that of tutorials. At quite some distance from the above, the e-resources of downloadable sample code, blogs, and e-books, although ranking in the order shown here, were comparable amongst themselves in terms of preference ratings. Similarly, the e-resources of online quizzes, e-peer support via developer forums, textbook websites, and e-peer support via social networking forums, although ranking in the order shown here, formed a relatively uniform profile near the bottom of the student preferences. Finally, CBT's were shown to be a distant last in the ranking scheme.

When asked to rate specific e-resources in terms of perceived usefulness in solving programming problems (i.e. Java coursework assignments), students once again showed a strong leaning towards tutorials, with the three related subcategories as the most preferred in the following order: online tutorials (text and images), video tutorials (screen-cams and how-to), and video tutorials (recorded theory lectures). This ranking, which places tutorials at the top of preference rating, closely mirrors that pertaining to the usage of the same resources for studying Java. Ratings of "very useful" to "useful" were given by 10 of the respondents to video tutorials (screen-cams and how-to's) and online tutorials. Blackboard again rated as the next most popular e-resource, although it's perceived strength was not in the same proximity as that of tutorials, and was followed quite closely by blogs, e-books, and downloadable sample source code, in that order. At quite some distance from the above, the e-resources of textbook websites, e-peer support via developer forums, and online quizzes, although ranking in the order shown here, formed a relatively uniform profile near the bottom of the student preferences. Finally, CBT's and e-peer support via social networking forums ranked extremely poorly as e-resources of choice.

The tables below display the perceived usefulness of learning resources by the number of students in their course of study and in their course of solving programming assignments.

Table 1

LEARNING RESOURCES	**USE TO STUDY** (out of 15 students surveyed)
Lab Exercises (hands-on Practice)	9
Lectures	8
Video Tutorials: Screen-cams and how-tos (more practical hands-on examples)	7
Video Tutorials: Live Recorded Lectures (theory)	6
Class Notes	5
Discussion and/or study with classmates	5
Blackboard	3
Downloadable working sample source code	3
e-books	3
Textbook (programming examples/exercises)	2
Java Manual (IDE Help)	2
Online Tutorials (Text and Images)	2
Textbook Website	1
Textbook (theory)	0
Online Quizzes	0
Blogs	0
e-peer support through Developer Forums	0
e-peer support through Social Networking web sites (like Facebook)	0
CBTs	0

Table 2

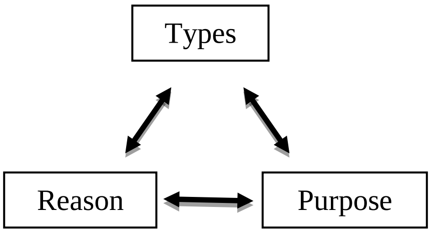
LEARNING RESOURCES	**USE TO SOLVE PROBLEMS** (out of 15 students surveyed)
Class Notes	9
Online Tutorials (Text and Images)	8
Discussion and/or study with classmates	7
Textbook (programming examples/exercises)	6
Video Tutorials: Screen-cams and how-tos (more practical hands-on examples)	5
Video Tutorials: Live Recorded Lectures (theory)	5
Java Manual (IDE Help)	4
Textbook (theory)	4
e-books	3
Textbook Website	3
Blogs	3
Lectures	2
Lab Exercises (hands-on Practice)	1
Blackboard	0
Downloadable working sample source code	0
Online Quizzes	0
e-peer support through Developer Forums	0
e-peer support through Social Networking web sites (like	0

Facebook)	
CBTs	0

Having identified key resources perceived as helpful by the students, the semi-structured interviews were conducted to explore the reasons as well as the selection criteria applied by the students in order to use a learning resource.

The initial theoretical framework (figure 2) shows that there are relationships between the types of resources used by students, why these resources are used and how.

Fig. 2

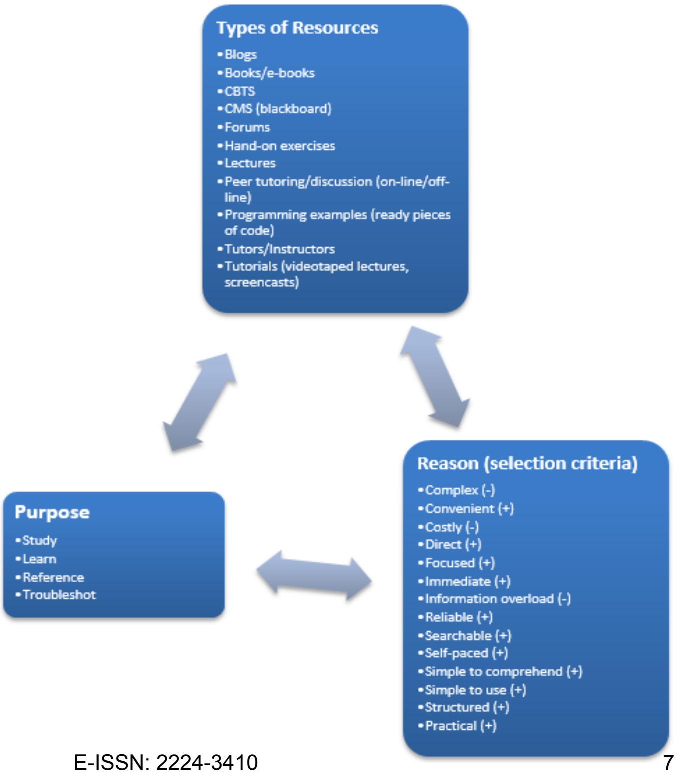


The list of purposes was enhanced with new key sub-categories that emerged from the interviews.

- Process of Studying
- Process of Learning
- Process Referencing (syntax)
- Process of Solving programming Exercises/Troubleshooting

The interviews also generated the list of reasons (positive and negative) for adopting learning resources:

● Complex (-)	● Convenient (+)
● Costly (-)	● Direct (+)
● Focused (+)	● Immediate (+)
● Information overload (-)	● Reliable (+)
● Searchable (+)	● Self-paced (+)
● Simple to comprehend (+)	● Simple to use (+)
● Structured (+)	● Practical (+)



As can be seen by the above elements students mostly characterize their resources based on positive feelings instead of negative. After the completion of the concept lists, the theoretical framework was enhanced to depict (fig.3)

The elements where grouped according to the meaning/description as discussed by the students. The following interview extracts demonstrate the process of memoing during and after the interviews and how the main concepts where formulated.

Researcher: ….Do you study from the textbook?

Participant[a]: No, I find textbooks about programming boring… I find them very hard to follow and study… too much theory between the code… With tutorials the goals are clear and short, more focused. A textbook chapter starts by redefining the world… But if I had an electronic version of our textbook I would use it to search and locate specific concepts.

Focused (+)	Short and to the point Clear goals
Information overload (-)	Too much theory, too much information
Searchable (+)	Locate fast the information needed

…

Researcher: ….As you know I am performing an exploratory study on the types of learning resources students at your college use to learn programming. So the first question I would like to ask you is: which resources do you use to study for your JAVA course….

Participant[b]: I use lecture notes and instructor notes posted on blackboard…

Researcher: why?

Participant[b]: …because they are structured. They contain all the information I need to know for the course.

Researcher: Do you study from the textbook?

Participant[b]: No, with the exception of C language, in which I study "the bible"… written by the person who created the language itself, No never. I never study programming from textbooks. They are boring, and they cost too much…

…

Costly (-)	
Structured (+)	

…

Participant [e]: Online tutorials… I think they are the best…

Researcher: Can you expand a little more on that…?

Participant [e]: I like watching someone perform (write a program) while he explains, and I try to do it, either at the same time by pausing the video and doing and playing and pausing and doing… until I understand it, or watch all once and try to understand the concept and do it on my own. What I love about tutorials is that I can go back and re-view something I did not quite get in the first time… What I also like is that they are short and to the point, with clear objectives.

Researcher: any other resource that has helped you learn?

Participant [e]: Do friends count?

Researcher: Yes, of course…

Participant [e]: I speak to my classmates and I ask them how they solved this problem… I take input from multiple classmates and compare their solutions to mine… Or we sit all together and we try to make it work… The input, which I receive from classmates, is simpler and easier to understand…

….

Self-paced (+)	
Simple to comprehend (+)	Uses Simple words, simple concepts, not complicated
Focused (+)	Short and to the point Clear goals

The key concepts that emerged from the interviews are summarized in the following table.

Table 3

ELEMENT	MEANING/DESCRIPTION (MEMO)
Complex (-)	Difficult
Convenient (+)	Easy to use and access
Costly (-)	
Direct (+)	Immediate feedback, immediate solution
Focused (+)	Short and to the point Clear goals
Information overload (-)	Too much theory, too much information
Reliable (+)	Trust the source
Searchable (+)	Locate fast the information needed
Self-paced (+)	
Simple to comprehend (+)	Uses Simple words, simple concepts, not complicated
Structured (+)	
Practical (+)	Learn by doings, hands-on

4 Findings

Students use lectures and perform programming exercises (homework) in order to acquire knowledge and test this knowledge. On the other hand, students use forums and interact with their peers to exchange ideas, use lab practices, e-books and the language web site as a reference, and use textbooks, lab practices, class notes and tutorials to study. The purpose of this research though is to define possible relationships between the means and the reason.

If we triangulate the results from the survey, lab exercises appear to be the mostly highly preferred resource for studying and referencing we can conclude that for this specific group of students practicality and reliability are among the major factors behind their decision to use the particular resource. Students identified as practical most of the recourses they use. Tutorials as well as peer support fall in the category of “simple to comprehend” which is another popular selection criterion. Lastly, direct communication and immediate feedback is another key reason for the adoption or not of a resource. Interestingly enough, this group of students does not use their textbook to study nor to address problems.

Table 3

Resource Type	Purpose	Reason
Lab Exercises (hands-on Practice)	Study Reference	Practical Reliable
Lectures	Learn	Direct
Video Tutorials: Screen-cams and how-tos (more practical hands-on examples)	Learn Study	Simple to comprehend Focused Convenient Practical Self-paced
Class Notes	Study	Reliable
Discussion and/or study with classmates	Solve programming problems	Direct Simple to comprehend
Textbook programming	Learn Study	Practical

exercises		
Peers	Solve programming problems	Simple Direct Simple to comprehend

The purpose-category items are in line with the deep and/or surface approaches to learning. *"the deep approach was associated with a holistic style and intrinsic motivation (interest in the subject matter itself) to form a meaning orientation. Surface approach went with serialist style (a narrow, cautious stance relying on evidence and logical analysis) and fear of failure within are producing orientation, while strategic approach indicated a use of both deep and surface approach supported by a competitive form of motivation (need for achievement) combined with vocational motivation within an achieving orientation."* (Entwistle & Tait, 1990, p. 171)

While studying/learning belong to a deep approach to learning, reference and solving problems can be considered as parts of a surface approach. Relevant research on teaching and learning how to program has shown that learning a programming language requires a student to deploy both a deep and a surface approach to learning. This means that, although a programming language can be memorized, this is not a mandatory characteristic to master the programming language. In order to be efficient and proficient in programming, the student should learn how to think in computer terms (construct algorithms) and know where to look for "surface" information such as syntax rules.

All student participants in the interviews agreed upon the fact that "only the basics of a programming language can be taught", while the rest depends on practice and guided direction. They supported that they prefer to have control over the learning process. *"Perceived control refers to user perceptions about who controls the information retrieval process."* (Parikh & Verma, 1009)

The literature review on learning theories suggests that there are three widely accepted paradigms of learning: cognitive, constructivist and social.

The cognitive learning paradigm is based upon the view that the learner, rather than being a mere receptor of information or knowledge, is an assimilator of knowledge, with heightened understanding resulting from the capacity to recognize and interpret conceptual interdependencies and thus contextually enrich the knowledge acquired. The understanding of cognitive processes is essential to this paradigm, as related mental activities must be identified and targeted to promote the most effective learning. The learner's schema is viewed as an organized knowledge structure (Bruner, 1990) (Gagne, Yekovich, & Yekovich, 1993)

The constructivist-learning paradigm is based upon the view that a learner is capable of contracting his or her own knowledge, albeit within the framework of a subjective model of representation. This paradigm approaches learning as a process in which one integrates new information with previous knowledge and experiences (Duffy, Lowyck, & Jonassen, 1993) in order to actively construct an extended knowledge schema in a piece-wise fashion. (Steffe & Gale, 1995) Learning processes center around problem solving within loosely structured realms in order to promote self-realization and allow learners to adapt their mental models to newly discovered knowledge.

The socially situated learning paradigm is based upon the view that, above and beyond the strict and rigid confines of constructivism, a social dimension is intrinsic to the learning process. This paradigm considers the social environment to be that in which knowledge exists and throughout which it can be effectively disseminated. (Vygotsy, 1978; Hadjerrouit, 2007) As such, learners enhance, challenge, validate, and ultimately deepen their knowledge within the context of peer- or group-related activities involving communication, synergy, sharing, and overall interaction with others.

Mayers and Fowler (1999) indicated that learning is an iterative process (a cycle) that repetitively goes through three stages: conceptualization, construction and dialogue. The learning model proposed is a one of gradual refinement of understanding. These three stages of the learning model are in accordance with the learning paradigms. The Conceptualization phase is based on cognitive theory, because it focuses on organizing concepts and their relationships. The Construction phase is based on the constructivist theory, as it targets the creation of new knowledge through practice and problem solving. Finally, the Dialogue phase is concerned with peer collaboration and group discussion and is in line with socially situated learning.

The student-participants in this study show a strong preference for learning resources identified as practical, which makes their perception of learning programming to follow a constructivist theory.

A related study on the "use of learning resources by students when learning to program" (Eckerdal, 2006) identified some learning resources as being used in a search-for-meaning ways, while others in a superficial way. Search-for-meaning approach relates to what students identified as learning/studying resources, whereas superficial relates to reference or debug (solve problems) resources.

5 Conclusions

It should be noted that this study was conducted in a small college in Greece, and that the researcher is aware of the contextual factors underpinning data collection (age group, ethnicity, level of expertise, competence in using Internet resources, etc). The results should not thus be generalized to all undergraduate students learning how to program.

Additionally, of the 15 students surveyed and 8 interviewed, there was only one female student. As a result, the sample over-represents the male population. Another constraint underlying this research is that all participants were current students of the researcher, a fact that may have compromised the results due to impartiality.

Another constraint that should be noted is that, due to time limitations, the researcher did not have the opportunity to re-enter the field to perform further theoretical sampling, and the saturation of the categories might not have been reached.

General consensus amongst educators is that computer programming is a challenging endeavor, given that it involves a range of acquired skills and cannot be mastered using a strictly knowledge-based approach. As long as the Internet is considered to be a preferred technology to improve instruction in higher education (McArthur & Lewis, 1998), it is imperative that educators familiarize themselves with student perceptions as to which of the resources available are more useful.

It can be concluded from the findings that students are generally adverse to studying from textbooks, primarily because they are perceived as being overly theoretical, complex, and containing “too much information”. Student are instead committed to utilizing learning resources that they consider to be practical, focused, immediate, reliable/trustworthy, and easy to understand. The resources that seem to demonstrate these desired features are: lab exercises, lectures, online tutorials and peer support. The study of the impact the use of these resources has on student learning is beyond the scope of this study.

A key finding in Hadjerrouit’s (2007) study was that online resources play a major/positive role in influencing students during the process of learning how to program.

As computer programming is considered a discipline in which acquired skills, rather than discrete sets of knowledge, are called upon, it has become increasingly clear that the teaching of programming can greatly benefit from the model described by Mayers and Fowler (1999), in which learners first organize concepts and formulate relationships, then actively construct knowledge bases, and exchange ideas with their peers. Students seem to recognize and favor the directness, timeliness, efficiency, and effectiveness of socially situated learning. The World Wide Web also complements such a paradigm by extending the classical social context with a valuable and dynamic virtual dimension.

6 Reflections from the study

We found the project to be challenging and especially enlightening as to how a particular methodology can be applied to step through the process of investigating a research topic. It also enhanced our understanding as to how a researcher would go about formulating research questions and comparing and contrasting available methodologies.

In addition, the topic we covered in this paper is of particular interest to us as tutors, as the findings will help us to tailor our lectures by focusing on e-resources that are considered most effective by the students themselves. It will also serve as the basis for future research into whether the usage of such resources actually has an impact on students’ academic performance.

Finally, we should note that time constraints on this project were quite pressing. This limited our ability to expand the data collection and deepen the analysis of the results, as well as triangulate the findings with additional relevant literature.

References:

[I] Bruner, J. (1990). *Acts of meaning.* Cambridge, MA: Harvard University Press.

[II] Coyne, I. T. (1997). Sampling in qualitative research. Purposeful and theoretical sampling; merging or clear boundaries? *Journal of Advanced Nursing* (26), 623-630.

[III] Creswel, J. W. (2003). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches* (2nd ed.). London: Sage Publications Ltd.

[IV] Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly , 12* (3), 319-340.

[V] Duffy, T. M., Lowyck, J., & Jonassen, D. H. (1993). *Designing environments for* *constructive learning.* Berlin: Springer-Verlag.

[VI] Eckerdal, A. (2006). *Novice Students' Learning of Object-Oriented Programming.* Uppsala University, IT Department.

[VII] Entwistle, N., & Tait, H. (1990). Approaches to Learning, Evaluations of Teaching, and Preferences for Contrasting Academic Environments. *Higher Education , 19* (2), 169-194.

[VIII] Fowler, L., Allen, M., Armarego, J., & Mackenzie, J. (2000). Learning styles and CASE tools in Software Engineering. *Teaching and Learning Forum.*

[IX] Gagne, E., Yekovich, C., & Yekovich, F. (1993). *Cognitive psychology of school learning.* New York, NY: HarperCollins.

[X] Glaser, B. (1992). *Basics of Grounded Theory Analysis.* Mill Valley, California: Sociology Press.

[XI] Glaser, B. (1994). Theoretical Elaboration of Quantitative Data. In B. Glaser, *More Gounded Theory Methodology: A Reader* (pp. 197-232). Mill Valley, CA: Sociology Press.

[XII] Glaser, B. (1978). *Theoretical Sensitivity.* Mill Valley, CA: Sociological Press.

[XIII] Glaser, B., & Strauss, A. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research.* Chicago: Aldine Publishing Company.

[XIV] Hadjerrouit, S. (2007). A Blended Learning Model in Java Programming: A Design-Based Research Approach. *Computer Science and IT Education.*

[XV] Jones, M. L., Kriflik, G., & Zanko, M. (2005). *Grounded Theory: A theoretical and practical application in the Australian Film Industry.* Retrieved from University of Wollongong: http://ro.uow.edu.au/commpapers/46

[XVI] Krauss, S. (2005). Research Paradigms and Meaning Making: A Primers. *The Qualitative Report , 10* (4), 758-770.

[XVII] Lösch, A. (2003). Grounded Theory for Researching. *56*, pp. 133-144. Libri.

[XVIII] Mayers, J. T., & Fowler, C. J. (1999). Learning technology and usuability: A framework for understanding courseware. *Interacting with computers* (11), 485-497.

[XIX] McArthur, D. J., & Lewis, M. W. (1998). *Untangling the Web: Applications of the Internet and Other Information Technologies to Higher Learning.* Rand Corporation, The.

[XX] Morse, J. M. (1991). In Qualitative Nursing Research: A Contemporary Dialogue. In *Strategies for sampling.* (pp. 127-145). Newburry Park, California: Sage Pubications Ltd.

[XXI] Parikh, M., & Verma, S. (2002). Utilizing Internet technologies to support learning : an empirical analysis. *International Journal of Information Management , 22*, 27-46.

[XXII] Robson, C. (2003). *Read World: Research: A Resource for Social Scientists and Practitioner-Researchers.* Offord: Blackwell Publishers Ltd.

[XXIII] Sandelowski, M., Holditch-Davis, D., & Harris, G. B. (1992). Using qualitative and quantitative methods: the transition to parent-hood of infertile couples. In *In Qualitative Methods in Family Research* (pp. 301-323). Newbury Park, California: Sage.

[XXIV] Seibold, C. (2002). The place of theory and the development of a theoretical framework in a qualitative study. *Qualitative Research Journal , 2*, 3-15.

[XXV] Selvaraj, N., & Fields, B. (2009). A Grounded Theory Approach towards Conceptualizing CIS for Heterogeneous Work Communities. *British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (pp. 471-479). British Computer Society.

[XXVI] Steffe, L. P., & Gale, J. (1995). *Constructivism in education.* New Jersey, NJ: Lawrence Erlbaum Associates.

[XXVII] Strauss, A., & Corbin, J. M. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory.* London: Sage Pubications Ltd.

[XXVIII] Vygotsy, L. S. (1978). *Mind in society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press.

[XXIX] Yin, R. (1994). Designing case studies. In R. Yin, *Case study research: design and methods* (pp. 18-53). London: Sage Pubications Ltd.