CHALLENGES IN LEARNING PROGRAMMING DIALECTS FOR UNDERSTUDIES

[1]Mr. Varun S. Moruse, [2]Ms. Supriya A. Chougule
[1]Assistant Professor, [2]Assistant Professor
[1]Computer Science and Engineering,
[1]Dr.D.Y.Patil Pratishthan's College of Engineering, Kolhapur
[2]Department of Accountancy
[2]D.R.K. College of Commerce, Kolhapur

Abstract*—Due to the exponential development of advances in programming, one of the key aptitudes recognized as an essential education for what's to come is programming. Writing computer programs is a critical fundamental expertise for software engineering understudies. Most understudies discovered it is difficult to learn and score a decent. This paper discusses the study to recognize issues and causes looked by programming understudies. The survey was directed at the various Engineering colleges under Shivaji University, Kolhapur. Information was gathered from 126 understudies who addressed on the web poll. The present study recognized a few noteworthy issues; reference materials, addresses and lab session approaches, issue unraveling capacity, time administration and self-assurance. The discoveries will be utilized to enhance the courses and in this manner guarantees the acknowledgment obviously results, diminish the disappointment rates and increment understudies' enthusiasm for programming

***Index Terms*— development, programming, issues, fundamental, understudies.**

I. INTRODUCTION

Writing computer programs is a piece of the essential educational modules for software engineering and data innovation. The capacity to comprehend and actualize the programming dialect is vital expertise for the understudies to obtain. The programming dialect subjects are essential for the other programming courses offered at the third and subsequently semester. The subjects are shown utilizing programming dialect and directed by address (hypothesis) and useful exercise in the research facility. This paper discusses about the investigation to recognize the understudies' issues in the subject of programming and reasons for those issues. The examination additionally researches whether the reasons for the issues control understudy execution in this subject. Understudy execution is measured from their examination review.

The quantity of Computer Science understudies who couldn't write error-free computer programs and the failure rates in programming courses at the college level are pointers that figuring out how to program is a troublesome undertaking. One source proposes that the disappointment rate is as high as 30 percent (Guzdial and Soloway, 2002). Understudy's success or failure in programming courses ordinarily influences the choice whether to proceed in the field of Computer Science or not. Sometimes understudies were made a request to change their courses to a less entrusting one because of an excessive amount of disappointment in a specific field of study. On the off chance that an understudy comes up short, or passes with a struggle, that understudy isn't probably going to enlist for a take after on course in that field and they invest more energy in different courses to get a decent aggregate gross item normal which tends to influence their ability in the field of concern. Disregarding inquires about on factors that impact the enrolment and accomplishment of understudies in programming, it is as yet not completely comprehended what makes PC programming simple for a few, however troublesome and disappointing for others.

II. LITERATURE REVIEW

Writing computer programs is a subject that included ability in planning calculation, composing program, understanding the language structure and also the rationale of the program. Writing computer programs is among the hardest subject to learn because of a few reasons. (Daly, 1999) (Jenkins, 2002).

Writing computer programs is another subject for some understudies. We can't anticipate that understudies will be gifted in programming inside 14 weeks. Be that as it may it might took 10 years for somebody to be talented software engineer (Winslow, 1996).

Understudies' learning style is additionally unique for every understudy. Some like exchange with companions while different likes to think about alone. Whatever their styles are yet the most essential is the way they considering. This is on account of figure out how to program included an alternate state of mind (Mayer, Dyck et al., 1986).

Educators assume a noticeable part in conveying the information proficiently to the understudies (Gomes and Mendes, 2007). Educators are capable to disclose to the understudies obviously and propose to them the answer for their issues. Learning condition like size of the class, schedule opening for the class, and length of the class additionally can influence understudies' inspiration (Jenkins, 2001). These can be an issue to the understudies if the educators neglect to deal with the circumstance other than the different learning styles of the understudies.

As indicated by a few analysts, understudies numerical capacities frequently decidedly ponder their programming capacities (Bennedsen, 2008), (Sauter, 1986). Subsequently, educators tend to plan educational programs that favors such understudies, while other viewpoint, for example, critical thinking may be ignored.

Scientists in (Hawi, 2010) accentuate ten factors that influenced their understudies, and some of those elements that we additionally saw through perception and meetings with our understudies were: "learning system", "absence of study", "absence of work on", "showing technique", "exam nervousness" and "conning". Some of those components were likewise noted in (Bennedsen, 2008) as indicators of progress for programming. Understudies have extra challenges with conceptual considering. In the exploration led by (Eckerdal, Thun, and Berglund, 2005), understudies were met with the reason for deciding whether they comprehended what realizing programming implies. Numerous understudies expressed it is exceptional state of mind, however were not ready to portray in detail.

Attitude, they say, is the way to progress. As indicated by Erin (2008), mentality is similarly as critical as capacity. (Erin N. Goodykoontz, 2008). Popham, (2005) propose that understudy states of mind toward a subject prompt scholarly achievement in that subject. Specialists have additionally discovered that self-viability is one of the components that impact understudies in procuring programming aptitudes. For example Askar and Davenport (2009) take a shot at examination of components identified with self viability for Java programming demonstrated that self-adequacy among guys were more grounded than that of females. This may represent why male command the programming scene. Miliszewska and Tan (2007) deduced in their work on difficulties of giving programming abilities on understudies that good laboratory facilities, joint effort and on-line accommodation frameworks were among the components that assume vital parts. The educational modules association and the showing techniques were additionally observed to be factors that influence understudies' execution in programming, as indicated by Tavares et al (2001).

III. Research Objectives

This study aims to explore the views of students and the difficulties they experience while learning Fundamental Programming courses.

In particular, the objectives of the study are as follows:

To study the level of understanding and attitude of students studying programming.
To investigate the pattern of study and difficulties in learning programming.
To explore the reasons behind disliking programming.
To find the causes of poor performance in programming among students.
To give suggestions and provide recommendations for increasing interest of students n Programming subject.

IV. Methodology

A set of questionnaires was developed as an instrument for this research which includes six main sections:
(i) Section A: Respondent profile & Opinion of students on programming studies,
(ii) Section B: Feedback about PL teachers,
(iii) Section C: Reasons behind disliking Programming
(iv) Section D: Opinion about teaching/learning pattern

Section A covers information about respondent's background and their programming experience. Section B asks about the programming subject teacher experience. Teaching methods and course delivery by lecturers and teachers are questioned in this section. Section C is related to the various reasons behind disliking programming. Section D focuses on learning methods, revision methods, handling of assignment and time management. However, this paper only focuses on the problem potentially faced by students as well as the source of problems .More importantly, this paper will also discuss the relationship between the problems and student performance either directly or indirectly. This questionnaire was distributed online to all students taking the programming subject. Due to the Time limitation convenient method was used for sample selection. In total 126 responses were collected for the research study purpose. The samples were selected by using convenient sampling method.

V. Discussion

In the light of the consequences of the examination performed on inquiries to research significant issues in programming, the investigation found that understudies don't have abnormal state of certainty to perform singular assignments and rely upon each other. Understudies activity is likewise exceptionally negligible on the grounds that a large portion of them, especially understudies with direct and powerless execution dependably seek after assistance from different sources, for example, the appropriate response conspire from reference books, and help from speakers, educators and companions to enable them to tackle specific activities or undertakings. More critically than that, understudy's exertion in acquiring materials for addresses and research center sessions is still at the very least level. The greater part of understudies still depends totally on notes or slides arranged by teachers despite the fact that they know that these notes are insufficient to help their comprehension. There are likewise understudies who utilize other reference sources however the numbers are too little. The investigation additionally discovered understudies who performed incredibly utilize different contrasting options to build their insight by getting notes from senior companions and different references from the Internet. Furthermore, with bunches of time spent on addresses or lab sessions for every one of the subjects taken, it is exceptionally troublesome for understudies to do amendments and additional activities to upgrade their programming aptitudes. Substantial number of understudies in a single class may likewise influence the concentration of the teacher and understudies. The circumstance makes it troublesome for instructors and educators to recognize understudies who are powerless in essential programming abilities. Among the proposed changes that should be possible is to utilize the combine programming technique as an option, keeping in mind the end goal to enable understudies to take care of their programming issues all the more usefully. This strategy can likewise enable understudies to compose programs with more elevated amount of certainty than doing them independently. In a roundabout way, the issue of sharing answers can be diminished. Utilization of gatherings as a medium ought to be more energized, so understudies don't feel humiliated to ask the speakers or instructors up close and personal. This strategy could be useful for direct and powerless understudies since they at times feel uncertain of what sort of the inquiry should they inquire.

VI. Analysis and Interpretation of Data

The collected data was analyzed by using percentage and interpreted thereon to derive meaningful conclusion.

Table 1- Responses of the student on learning experience of Programming

Number of Students (Total)	**Experience**				
	Less than 6 months	***Up to 1 Year***	***Between 1 to 2 Years***	***Between 2 to 3 Years***	***Between 3 to 4 Years***
126	15	11	40	37	23

Interpretation:

It is clear from the table 1 that 15 students have less than 6 months experience of learning Programming and 11 had up to 1 year and 40 students had between 1 to 2 year and 37 had between 2 to 3 year and 23 had between 3 to 4 years learning experience of programming. Total 126 student's responses have been observed under the research study.

Table 2 - Opinion of the Students about Programming studies

	Opinion on programming studies	**Strongly agree**	**Agree**	**Neither agree nor disagree**	**Disagree**	**Strongly disagree**
1	I love to practice programming daily.	23.0	46.8	24.6	5.6	0.0
2	My basic Programming Concepts are clear.	11.9	61.9	18.3	7.9	0.0
3	My background in Mathematics is strong.	23.8	46.8	20.6	7.1	1.6
4	Programming doesn't excite me.	7.9	18.3	36.5	22.2	15.1
5	I lose my patience while programming.	9.5	27.0	31.7	19.8	11.9
6	I'm not a "details person"	6.3	22.2	41.3	19.8	10.3
7	I cannot give enough time to apply what I have learned.	17.5	46.0	20.6	10.3	5.6
8	I want to stay in the comfort zone	19.8	29.4	25.4	15.9	9.5

Interpretation:

In the Table 2 on programming studies, it is observed that 69.8% students agreed that they love to practice programming daily and only 5.6% students admitted that they dislike practicing. 70.6% understudies have arithmetic solid foundation. And 37.3% understudies differ that programming doesn't energize them and 36.5% understudies conceded unbiased reaction on the same. 41.3% understudies are unbiased about points of interest programming. Most noteworthy number of understudies i.e. 63.5% conceded that they can't give enough time for executing programming ponders. Just about half understudies wish to be in safe place that demonstrates their inclination of protection from acknowledge new things.

Table 3 Feedback about Teachers

	Feedback about teachers	**Strongly agree**	**Agree**	**Neither agree nor disagree**	**Disagree**	**Strongly disagree**
1	My PL teachers develop interest in me about programming.	25.4	41.3	15.9	11.1	6.3
2	I think My PL teachers fail to do fair assessment.	6.3	26.2	32.5	22.2	12.7
3	My PL teacher asks me to do unethical programming (like: copy readymade codes).	7.1	15.1	22.2	28.6	27.0
4	My PL teachers waste time on irrelevant matters in class.	10.3	10.3	23.8	29.4	26.2
5	My PL teachers keep me informed about updated concepts.	25.4	38.9	21.4	7.1	7.1
6	My PL teachers attend to me whenever I have difficulty with their course.	29.4	44.4	20.6	4.0	1.6
7	My PL teachers use advanced teaching methods and techniques.	22.2	35.7	30.2	7.1	4.8
8	My PL teachers have good knowledge about programming.	38.1	40.5	13.5	6.3	1.6

9	The institute has Positive Environment for studies.	22.2	49.2	20.6	5.6	2.4
10	The institute arranges workshops about programming languages	11.9	38.9	30.2	13.5	5.6

Interpretation:

Dominant part of understudies concurred that their PL educators create enthusiasm about programming. 35% understudies are sure about reasonable evaluation. 55.6% understudies reacted that their PL instructor never makes a request to do unscrupulous programming. A large portion of the understudies concurred that PL instructors keep them educated about refreshed ideas. Around 75% understudies reacted that their PL educators fathom their challenges about course.

More prominent number of positive reaction has been recorded about utilization of cutting edge showing strategies, systems and information of instructors. Most elevated reaction was conceded about foundation with positive examination condition and their drives.

Table 4 - Reasons behind disliking Programming

	Reasons behind disliking Programming	**Strongly agree**	**Agree**	**Neither agree nor disagree**	**Disagree**	**Strongly disagree**
1	It's tedious task & it's difficult to code.	7.1	30.2	29.4	19.8	13.5
2	It's physically painful process as I need to seat for hours.	4.0	15.1	31.0	32.5	17.5
3	I get nervous when I get error message while coding.	8.7	27.0	31.7	17.5	15.1
4	I hate Logic Problems.	4.0	15.9	32.5	30.2	17.5
5	It takes more time while programming than I have expected.	11.9	34.9	34.1	11.9	7.1
6	It is frustrating that every other language has its own syntax.	10.3	34.9	28.6	15.9	10.3
7	The more focus is given on written assignments.	23.0	30.2	22.2	12.7	11.9
8	I don't understand the subject taught in classroom teaching.	5.6	23.0	25.4	27.8	18.3
9	I have been forced by my parents to enroll for this course.	3.2	3.2	24.6	27.8	41.3
10	Attitude of my PL teacher towards subject is not good.	7.1	17.5	24.6	23.8	27.0
11	I'm not sure about employment opportunities in programming.	8.7	18.3	32.5	23.8	16.7

Interpretation:

Majority of students responded that programming is tedious task & they find it difficult to code. Most of them have disagreed that it is physically painful process. High number of students love the logic programming. Most of the students have agreed that they feel frustrated that every other language has its own syntax. Highest number of responses is recorded about more focus is given on written assignments. Majority of students responded that they enrolled to the respective courses by their willingness. Large number of students has responded that they are sure about employment opportunities.

Table 5 Opinion about teaching learning Pattern

	Opinion about teaching/learning pattern	**Strongly agree**	**Agree**	**Neither agree nor disagree**	**Disagree**	**Strongly disagree**
1	Higher percentage of copying readymade codes in project work.	10.3	28.6	23.8	23.0	14.3
2	Syllabus contents are outdated technologies.	24.6	33.3	25.4	13.5	3.2
3	Non availability of proper study material.	14.3	31.7	21.4	27.0	5.6
4	PL teachers are well trained.	29.4	38.1	20.6	7.1	4.8
5	My PL teacher fails to complete Syllabus within period.	12.7	21.4	30.2	19.0	16.7

Interpretation:

About portion of the understudies concurred that there is higher level of duplicating readymade codes in venture work. Greater part of understudies reacted that syllabus substance are obsolete advances. Expansive number of understudies concurred that there is inaccessibility of concentrate material. Most astounding number of understudies concurred that their PL educators are all around prepared.

VII. Suggestions

Characterize Learning Goals, Increase Learning Interest and Improve Study Motivation

Since the students nearly have no understanding of program plan, it isn't important to clarify the solid showing substance in the primary lesson. The emphasis of the top of the line is to present the significance of the course. Possibly we can begin the main lesson by running a little scale game, and the instructor can tell the understudies that the diversion can be modified utilizing C language. So understudies can know the part of this course well. Amid the procedure of study, understudy may meet troubles, however when they review the parts of the programming plan, they will continue considering the course rapidly. Instructors may add new learning to keep up the propelled idea of the educating content.

Pick the Proper Textbook According to Student Character

Students read number of programming books The weakness is that understudies may recall the idea however they cannot join all the information with each other, and understudies cannot comprehend some solid inquiries themselves. The comprehend level is not the same as each other and all understudies are bad at comprehension. So we should pick the reading material with the character of use.

Change Teaching Mode and Emphasize the Different Contents

The method of teaching consolidates board with chalk. Since the lesson of C program configuration is a course with the pragmatic character, the program exhibition cannot be appeared out on the writing board. The conventional showing ways cannot fulfill the necessities. We give the method for consolidated customary showing technique with the sight multimedia to fulfill a wide range of prerequisites. The acceptable outcomes can be accomplished. Amid the way toward instructing, distinctive inquiries which the understudies must be addressed can be set.

Orchestrate Task of Practice

We realize that a few understudies have no contemplating inspiration. On the off chance that educators don't organize the training undertaking, at that point much time will be squandered. So as to change the express, some programming assignments must be masterminded to the understudies. Utilizing the technique understudies may feel pressure. We will applaud the understudies who complete the errand well, empower the understudies who have intriguing in programming yet don't completed the undertaking admirably and reprimand the understudies who don't complete the assignment.

Reinforce Checking the Practical Ability

The examination will changed into a test which gives careful consideration to check the pragmatic capacity. In the event that the understudies don't have the foggiest idea about the essential hypothesis focuses they cannot get the higher score. So this sort of examination can lead the understudies to give careful consideration to programming.

VIII. Conclusion

As it was found in the study that majority of students practice daily. Students are still in dilemma whether programming excites them or not. Almost half of the students are not aware about the detailed programming. Most of the students don’t give enough time for programming. Almost 50% students wish to be in comfort zone that shows their tendency of resistance to accept new things.

Understudies that enlist undergrad programming courses are generally tenderfoots with no programming background. Since distinctive analysts indicate that numerical capacities influence programming as we expressed over, the connection examination was led and there is high connection between understudies' achievement in basic programming and numerical courses. Taking everything into account, these discoveries have recognized the issues exist in programming subjects among understudies and their causes. Changes in the technique for directing these courses must be a high need and ought to be improved every now and then. Speakers and instructors can utilize elective techniques to pull in understudies and help increment the interests in the realm of programming. Understudies ought to be more arranged with gaining notes before class, brief perusing the themes to be instructed, taking note of down the vital focuses amid addresses, inquiring as to whether they don't comprehend the hypothesis, inquiring as to whether the punctuation of program if hard to comprehend amid lab sessions, doing extra activities and finding different sources or references. Understudies are additionally anticipated that would enhance their learning style and be more aggressive to put themselves in a gathering of understudies with magnificent execution.

References

[1] Daly, C. (1999). RoboProf and an introductory computer programming course. *SIGCSE Bull*. 31(3): 155-158.

[2] Bennedsen, J. (2008). Teaching and learning introductory programming: a model-based approach. Doctoral Doctoral dissertation, University of Oslo, Oslo.

[3] Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. Computers & Education, 54(4), 1127-1136.

[4] Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., et al. (2007). A survey of literature on the teaching of introductory programming. ACM SIGCSE Bulletin, 39(4), 204-223.

[5] Cohen, L., Manion, L., & Morrison, K. (2007). Research Methods in Education (6th Edition). London: Routledge Falmer.

[6] Clark, R.W., Threeton, M.D., & Ewing, J.C. (2010). The Potential of Experiential Learning Models and Practices In Career and Technical Education & Career and Technical Teacher Education. Journal of Career and Technical Education, Vol. 25, No. 2, Winter, 48-49

[7] Vaidyanathan, S. (2012, 09). Should Kids Learn to Code in Grade School? Mind Shift.

[8] Jenkins, T. (2001). The motivation of students of programming. Proceedings of the 6th annual conference on Innovation and technology in computer science education. Canterbury, United Kingdom, ACM: 53-56.

[9] Mayer, R.E. , Dyck, J. L., et al., (1986). Learning to program and learning to think: what's the connection? Commun. ACM 29(7): 605-610.

[10] Winslow, L.E. (1996). Programming pedagogy - a psychological overview. SIGCSE Bull. 28(3): 17-22.

[11] Sauter, V. L. (1986). Predicting computer programming skill. Computers & Education, 10(2), 299-302.

[12] Burleson, W. (2005). Developing creativity, motivation, and self-actualization with learning systems. *International Journal of Human-Computer Studies, 63*(4), 436-451.

[13] Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006). Not seeing the forest for the trees: novice programmers and the SOLO taxonomy. ACM SIGCSE Bulletin, 38(3), 118-122.

[14] Penny, K., Frankel, E., & Mothersill, G. (2012).Curriculum, climate and community: A Model for Experiential Learning In Higher Education. *International Technology, Education and Development Conference (INTED) Proceedings (6th),* 3-4

[15] Ritz, J. (2011). A Focus on Technological Literacy in Higher Education. *Journal of Technology Studies*, 31-40.

[16] Candy, L. (1997). Computers and creativity support: knowledge, visualisation and collaboration. *Knowledge-Based Systems*, *10*(1), 3-13.

[17] Gomes, A. and Mendes, A. J. (2007.) Learning to program - difficulties and solutions. *International Conference on Engineering Education* – ICEE 2007, Coimbra, Portugal.

[18] Shein, E. (2014, 02). Should Everybody Learn to Code? *Communications of the Association for Computing Machinery, 57*(2), pp. 16-18. Retrieved from Communications of the ACM.

[19] Eckerdal, A., Thun, M., & Berglund, A. (2005). What does it take to learn 'programming thinking'? Paper presented at the Proceedings of the first international workshop on Computing education research, Seattle, WA, USA.

[20] Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. ACM SIGCSE Bulletin, 37(3), 14-18.

[21] Jenkins, T. (2002.) On the difficulty of learning to program. In *3rd annual Conference of LTSN-ICS*.

[22] Crow, D. (2014, 02). Why Every Child Should Learn to Code. *The Guardian*.

[23] Henn, S. S. (2014, 01). *Computers Are The Future, But Does Everyone Need to Code?* Retrieved from All Teach Considered NPR:http://www.npr.org/blogs/alltechconsidered/2014/01/25/266162832/computers-are-the-future-but-does-everyone-need-to-code

[24] Lecownski, S. (2014, 10). Computer Programming in Design Education. (G. Contreras,

[25] DeLoura, M., & Paris, R. (2013, 12). *Don't Just Play on Your Phone, Program It.* Retrieved from Whitehouse.gov: http://www.whitehouse.gov/blog/2013/12/09/don-t-just-play-your-phone-program-it Interviewer)

[26] Dees, D.M., Ingram, A., Kovalik, C., & et al. (2007). A Transactional model of College Teaching. *International Journal of Teaching and Learning in Higher Education, Vol. 19, No. 2,131-133*

[27] Radenski, A. (2006). "Python first": a lab-based digital introduction to computer science. SIGCSE Bull., 38(3), 197-201.

[28] Bicer, S. (2012, 09). *Adapt or Die.* Retrieved from Business War Blogspot: http://businessaswar.blogspot.hk/2012/09/adapt-or-die.html

[29] Boden, M. A. (1998). Creativity and artificial intelligence. *Artificial Intelligence*, *103*(1), 347-356.

[30] Boden, M. A., & Edmonds, E. A. (2009). What is generative art? *Digital Creativity*, *20*(1-2), 21-46.

[31] Brinson, S. (2014, 10). *How to live a creative life that matters.* Retrieved from DIY Genius: http://www.diygenius.com/how-to-live-a-creativelife-that-matters/

[32] Chapple, C. (2013, 09). *UK Govt Outlines Computer Science Curriculum.* Retrieved from Develop: http://www.develop-online.net/news/uk-govtoutlines- computer-science-curriculum/0115853

[33] Cuni, B. (2014, 09). Computer Programming in Design Education. (G. Contreras, Interviewer)

[34] Loukides, M. (2014, 01 22). *The Reason Everyone Should Learn to Code.* Retrieved from Radar: http://radar.oreilly.com/2014/01/the-reasoneveryone- should-learn-to-code.html

[35] Marucs, G., & Davis, E. (2014, 06). *Do We Really Need to Learn to Code.* Retrieved from The New Yorker: http://www.newyorker.com/tech/elements/do-we-really-need-to-learn-to-code

[36] Pearce, K. (2013, 12). *Why You Should Learn to Code.* Retrieved from DIY Genius: http://www.diygenius.com/learn-to-code-online/

[37] Thornburg, D. (2013, 01). How Disruptive Technologies are Leading the Next Great Revolution. (S. Noonoo, Interviewer)

[38] Trovalds, L. (2014, 06). Linux creator Linus Trovalds: I do not believe everyone should learn to code. (D. Love, Interviewer) Retrieved from The Beat.com: http://venturebeat.com/2014/06/09/linux-creator-linus-torvalds-i-do-not-believe-everybody-should-learn-to-code/