

Analysis of learning environment and influence of environment among problem solvers

BY

**Mehedi Hasan
ID: 161-15-7068**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Mr. Saiful Islam
Senior Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

MAY, 2020

APPROVAL

This Project/internship titled **Analysis of learning environment and influence of environment among problem solvers** submitted by Mehedi Hasan, ID No: 161-15-7068 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 8th July 2020.

BOARD OF EXAMINERS



Dr. Syed Akhter Hossain
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Subhenur Latif
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

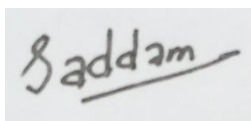
Internal Examiner



Raja Tariqul Hasan Tusher
Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Saddam Hossain
Assistant Professor

Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

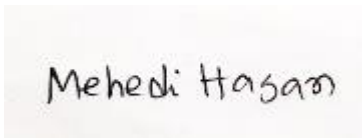
I hereby declare that this project has been done by me under the supervision of **Mr. Saiful Islam, Senior Lecturer, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Mr. Saiful Islam
Senior Lecturer
Department of CSE
Daffodil International University

Submitted by:



Mehedi Hasan
ID: 161-15-7068
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to Almighty Allah for His divine blessing makes it possible to complete the final year thesis successfully.

I am really grateful and wish my profound indebtedness to **Mr. Saiful Islam**, Senior Lecturer, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of *Natural Language Processing* helped me a lot to carry out this research. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Almighty Allah and Head, Department of CSE, for his kind help to finish my project and also to other faculty members and the staff of the CSE department of Daffodil International University.

I would like to thank **Mahmud Sajjad Abeer**(Alumnus, DIU) and **Mr. Ferduose Ahmed Foysal**(Alumnus and Lecturer, DIU) for their continuous help, support, and guidance from the beginning to end.

I would like to thank my entire course mates at Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

As we are growing up in the competitive world of science and technologies. Computer programmers are doing a great job overcoming the progress of science and technologies. The study includes the findings of the hard work, struggles, and back and forth of being a competitive programmer. We focus on the learning environment and the effects of the environment over programmers.

The methodologies we used for our research is quite simple. We used questionnaires to collect our qualitative and quantitative data to carry our study. Anaconda, a big framework of python is being used to clean, process, and analyze our data. We used some weighted value on each of the qualitative data fields to transform into quantitative data. Then we build up some relationship among the data field using both manual and scikit learn and matplotlib to visualize our findings.

What we found through the research is that every individual programmer has to go through a lot of hard work and struggles over the journey of being a professional programmer. They have to overcome a lot of weaknesses and pull themselves up from a lot of distractions from the environment. Sharing knowledge tendencies among them is very common and they found competitive programming helpful over their journey of being a programmer.

We do appreciate the hard work and struggle they have made through their journey and respect their sharing tendency. We should try to create the best learning environment and ensure the best support for them.

Table of Contents

APPROVAL	i
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
CHAPTER 1: INTRODUCTION	1-4
1.1 Introduction	1
1.2 Motivation	2
1.3 Rationale of the Study	2
1.4 Research Questions	2
1.5 Expected Outcome	3
1.6 Report Layout	3
CHAPTER 2: BACKGROUND	5-8
2.1 Introduction	5
2.2 Related Works	5
2.3 Research Summary	6
2.4 Scope of the problem	7
2.5 Challenges	7
CHAPTER 3: RESEARCH METHODOLOGY	8-12
3.1 Introduction	8
3.2 Research Subject and Instrumentation	8
3.3 Data Collection	8
3.4 Statistical Analysis	9
3.5 Detailed Methodology	9
3.6 Implementation Requirements	10
CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION	12-34
4.1 Introduction	12
4.2 Experimental Results	12
4.3 Descriptive Analysis	13
4.4 Comparison	29
4.5 Summary	29
CHAPTER 5: SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH	30
5.1 Summary of the Study	30

5.2	Conclusions	30
5.3	Recommendation	31
5.4	Implications for Further Study	32
	REFERENCES	33
	APPENDIX	34

List of Figures

Figure 4.1 Gender of respondents	13
Figure 4.2 Age of respondents	14
Figure 4.3 First hearing about competitive programming	15
Figure 4.4 Competitive Programming Experience	16
Figure 4.5.1 Daily average Practice	16
Figure 4.5.2 Time spent on practice vs Codeforces Rating	17
Figure 4.5.3 Time spent on practice vs Codeforces Rating	17
Figure 4.6 Time of practice	18
Figure 4.7.1 Learning Sources	19
Figure 4.7.2 Difference of having trainer, seniors, teammates	20
Figure 4.8 Solve count	21
Figure 4.9 CodeForces Ratings	21
Figure 4.10.1 Weaknesses in Problem-solving	22
Figure 4.10.2 Effect of weakness over progress	23
Figure 4.11.1 Distractions	23
Figure 4.11.2 Effect of distraction over progress	24
Figure 4.12.1 Reasons of doing Competitive Programming	24
Figure 4.12.2 Effect of mental happiness.	25
Figure 4.13 Sharing Tendencies	25
Figure 4.14 Academic Study vs Problem-solving	26
Figure 4.15 Residence	26
Figure 4.16 Financial State	27
Figure 4.17 Communication skill	28
Figure 4.18 Relay on Competitive Programming	28

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nowadays competitive programming has become very popular worldwide. Computer programming is now on its well-deserved hype and people all around the globe are getting involved in programming to solve real-life problems. That requires a good level of practice of problem-solving to come up with some solutions/applications/software that is highly efficient and dynamic.

This raises the need to develop good problem-solving communities and the arrangement of different programming contests. A programming contest is usually held on various online platforms hosted by educational organizations and supported by tech companies worldwide on a regular basis.

A programming contest involves participation of a good number of problem solvers to solve a bunch of real life, theoretical, logical and mathematical problems presented by the host. Usually, the scores are measured by the number of working solutions provided by the contestant and time taken to solve them under some given restriction(time limit, memory limit, output limit).

The programming contest acts as a backbone to create world-class programmers. It involves a lot of programmers to willingly participate and learn a lot of things related to programming and problem-solving.

For being a world-class programmer one must go through a lot of hard work, mental pressure, sufferings, and building up a passion for programming and problem-solving.

Success stories of programmers all around the world are basically the result of enormous support and hard-work where surroundings and environment play a great role.

Sadly, in Bangladesh, we've got minimal support yet they are doing good and representing Bangladesh to the world prestigiously. However, many of them don't get the proper support as we can see from our research.

1.2 Motivation

I had a great opportunity to be in one of the best programming communities in Bangladesh which is DIU's, Here I have gained a lot of experience. I have learned a lot and shared a lot from my learnings to others as well. But the path wasn't easy enough when I used to practice problem solving. We've to go through a lot of hard work and lack of necessary resources and support in many situations.

So I wish to find their pros and cons and bring them up and show the future community and authority some of the reflections.

I also wish to support the future community to build up their career and find problem-solving easy and more interesting.

1.3 Rationale of the Study

Newcomers face a lot of difficulties to overcome the first stage of learning Programming due to lack of knowledge, support, and guidance. Again, many Intermediate programmers get stuck in the middle and seek help. Even the advanced programmers look for a good mentor, trainer guide and a suitable learning environment.

While my goal is to minimize the struggle and let them see the truth. Also to encourage individuals and authority to support each other.

1.4 Research Questions

To identify the problem properly we would like to raise some questions. And try to find reasons and resolve them as we go.

- How are they doing well in such a complex domain of problem-solving?
- Why some of them can't do well in problem-solving?
- How will a newcomer cope with problem-solving and the community?
- How can we do better in the future?

1.5 Expected Outcome

In our research, we focus on the wellbeing of the problem solving and problem solvers community. What we want as an outcome is to be able to answer the raised questions with valid reasonings. Figuring out the reasons for the ups and downs of a programmer. Figuring out how to cope with some problems they are likely to arise in the path. Showing the authority if we need a lot of support from the community and authority. Also, finding ways to give problem solvers the required mental strength and previous experience of problem solvers to the newcomers so the path becomes easier.

1.6 Report Layout

The report is designed in a way such that readers will get a complete understanding of the subject we're focusing on.

Chapter 1 gives a basic understanding of the problem, the motivation behind it, and the objectives of this research.

In **Chapter 2** we're going to discuss the background, past related works, and challenges regarding this problem.

Chapter 3 will contain the back to the back explanation of the techniques and methodology used in this research.

Chapter 4 will mostly contain the analysis and comparison of results.

Chapter 5 will contain the summary, conclusion, constraints, and future scopes of the work.

CHAPTER 2

BACKGROUND

2.1 Introduction

As the problem arose people started to study it. Try to come with new solutions, resolve, modify, or upgrade to a newer one. Well, this chapter will discuss the related works with programmers, their learnings, their behaviors, their environments.

2.2 Related Works

There has been a lot of research about programmers on their behavior, tendencies, effect on environments, learning methodologies. All these researches take the understanding of programmers many steps ahead.

E. Soloway, J. C. Spohrer, 1989 [1] published a book “**Studying the Novice Programmer**” about how novice reason and solve problems within such complex domains. They shared the insight of improving programming instructions among them. They also tried to find the children's insight regarding programming. How they can be taught mathematics through computers. They also analyzed the development of programming ability and thinking skills among them and the methodology of learning to program and learning to think.

Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, Pekka Abrahamsson, 2018 [2] in their Journal “**What happens when software developers are (un)happy**” tries to find the Consequences of happiness and unhappiness among software developers.

Specifically, the positive consequences of happiness that are experienced related to the self and the negative consequences of unhappiness are experienced for external factors. What they have found is that the results of happiness and unhappiness are beneficial and detrimental for developers’ mental well-being, the software development process, and thus the produced artifacts.

Ming-Ten Tsai, Nai-Chang Cheng, 2010[3] in their research “**Programmer perceptions of knowledge-sharing behavior under social cognitive theory**” tries to find the key factors, including self-efficacy, expectancy theory, and organizational climate, on the software workers to intent to share knowledge, employing a social cognitive framework. And found that the knowledge sharing self-efficacy and outcome expectancy, as well as organizational climate will affect individual intentions to share knowledge. Additionally, organizational climate and perceived managerial incentives were found to positively encourage knowledge-sharing behavior.

Ruven E. Brooks, 1980[4] in his research “**Studying programmer behavior experimentally: the problems of proper methodology**” finds the application of behavioral or psychological techniques to the evaluation of programming languages and techniques is an approach which increased applicability over the past decade. He has concerned three major areas of methodological concern, the selection of subjects, materials, and measures are reviewed. The first two of these areas continue to present major difficulties for this type of research

Matthias M.Müller, 2007[5] in his research “**Do programmer pairs make different mistakes than solo programmers**” finds the Comparison of program defects caused by programmer pairs and solo developers. In his study, 42 programs developed by computer science students participated in an extreme programming lab course and they found that Programmer pairs make as many algorithmic mistakes as solo programmers but fewer expression mistakes than solo programmers and are available to a conclusion for straightforward problems. Pair programming seems to lead to fewer mistakes than solo programming.

2.3 Research Summary

The research mentioned above deals a lot with the reasoning and solving complex problems by novice programmers. Their thoughts on programming. Their skill up and learning methodologies. They also covered the behavior and knowledge sharing tendencies among them and mostly the effect of happiness over their career development and growth over product quality and tech

developments. Also a study shows that pair programming seems to lead to fewer mistakes than solo programming.

Aside from that there are some uncovered things about their performance, the effect of environments, their struggles behind learning, and support of the community.

In this research we're going to focus on different aspects than the above-mentioned works. Such as, how the newcomer can cope with such a complex domain and what the authority and individuals can do to do better in this field.

2.4 Scope of the problem

This research is going to put more value in the development of programmers at the root level. Previous research helps a lot to uphold the community a lot more. In this research we're going to focus on how programmers are doing well, why they can't, the effect of the environment on them, and what we could do to give the best support to them and also how a newcomer will cope with the community.

2.5 Challenges

The main challenge of this research is to work with the collected data. Mostly data is collected into mixed forms of Natural languages and numerical. We need to convert it into numerical with proper weighted values. Cleaning and processing the data in several phases of implementation. Again, working with numerals and finding answers to the raised questions. What we require is a huge collaboration between humans and machines.

Through the process we see a few more challenges and approaches to resolve them.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter discusses the methodology used to research the Analysis of learning environment and influence among problem solvers. The discussion will include the planning of the research, Research Subject and Instrumentation, Data Collection Procedure, Implementation requirements, and Statistical Analysis.

3.2 Research Subject and Instrumentation

Domain: We are doing the study over the learning environment and influences of the environment among problem-solvers. We will be focused on their learning procedures, their hard works, their weaknesses, distractions from the environment, and their progress through the study. Also, we will focus on finding the doings of individuals and authority.

Instrumentation: Well we are using questionnaires as an instrument for collecting the necessary data from every individual's respondent so that we can analyze and figure out our desired result. We are also using anaconda and python as an environment to analyze our data. We are using some python libraries Pandas, Openpyxl, Scikit-learn, Matplotlib to ingest, model, query, analyze, and visualize data.

3.3 Data Collection

Questionnaires are used to collect data from various local programming communities. The questionnaires covered four different areas each having sub-properties, but can be grouped as their workings with competitive programming, learnings, and supports, weaknesses and distractions, interests and benefits with competitive programming.

To make the research well balanced, collected data are divided into three major groups like Advanced, Intermediate, and Beginner category based on their performance and experiences.

3.4 Statistical Analysis

In our research, we will be working with both continuous and discrete data. We will be generating beautiful bar charts, pie charts, with the collected data. we will be able to find the statistics of each collected data field.

With the set of discrete data, we are going to plot them into various scattered plots. We will try to analyze the graphs and try to find the reasons.

3.5 Detailed Methodology

Well, now we will be discussing our proposed method for the study. Since we are studying over the data analysis of the learning environment and the influence of environment among problem solvers, we will be focus on the four main steps:

Descriptive Analysis: We will be describing the meaning of our collected data as a form of statistics. Every data collected from the questionnaires has a meaning within it and it's a big finding through the study.

Exploratory Analysis: We will find out some known and unknown relationships among the observed data fields.

Predictive Analysis: We will be covering some prediction over the data like (what will like to happen if!)

Prescriptive Analysis: We will utilize the art of technology and data practice in that part. We will try to implement models to analyze, predictions, and outcomes using machines including (machine learning algorithms and Artificial Intelligence).

With that in mind, we made wonderful questionnaires to collect the data to analyze. For creating the questionnaires, we had to go through a lot of basic questions about competitive programmers.

We mainly focus on three parts, their learnings, their progress, their weaknesses, and distraction from the environment.

We wanted to find the reasons why programmers do well, how they learn how often they sit for practice, is there any weakness for those we are not doing great. Is there anything pulling them off from the track? How much support we got from the community and authority?

Since we use google forms to collect data, google forms did a great job of finding the statistics for each individual's data responses.

But in our research, we need to find a relation with various data fields. For this, we had to organize, clean, and process data and imported it into a data frame to work on.

In some of the cases, we use raw python and pandas framework to process and analyze the data.

We use matplotlib to generate the charts and graphs and find relationships among them.

Unlike every research, we are trying to get something new, interesting, and productive from our research.

3.6 Implementation Requirements

In this section, we will go through some requirements to implement and analyze the study.

Hardware:

Processor:

Core i3 CPU @ 3.30GHz or above

RAM:

4.00 GB or above

Storage:

16 GB of Hard Disk Drive or above

Software Requirement:

Operating Systems:

- Windows OS – Windows 10

- Linux – Ubuntu 18.04 / Debian 10.0

Either of them should be fine

Environments:

- Anaconda
- Python 3.7 or above

Python Packages:

- Pandas
- Openpyxl
- Scikit-learn
- Matplotlib

Also, it must have internet connectivity available to function properly.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

Well, so far we have talked about the theoretical idea and implementation details of our research. In this section, we will be covering some experiments over our dataset. We will also provide an analysis and interpretation of the data field and experiments.

4.2 Experimental Results

Well, it's time to experiment with our data and validate what we claim in our research. In this section, we will discuss what we have found in our research and how it will be described in the descriptive analysis section.

First of all, we were wondering about how they are doing well in this complex domain of problem-solving,

What we find here is hard work is the key to success, we need to do a lot of hard work and spend a lot of time to practice, in the meantime we need to be guided to reach the desires we want. So there weren't any miracles or any shortcut to their success.

Secondly, we wanted to know why some of them are not doing well,

It's because of the failings of practice, lacking of not having a guide, or not having enough support.

Again it's because of their weakness and distractions from their environment we have some clear evidence that they are not giving excuses.

They just needed a healthy amount of supports to cope up.

We have claimed a how newcomer will cope with problem-solving and the community, So if we look at the data fields there are no shortcut or excuse to do so, But what we need is hardworking, patience, desire, admires and supports from the responsible to cope up with problem-solving and the community.

Finally, we have claimed how we can do a lot better in the future, again if we have a close look in the data field we can see that for an individual it's to do practice, have patience, and respect what he is doing. And for authority and community is to give the best supports they can.

4.3 Descriptive Analysis

As we go through, we will be describing each and every data field, its own effects, importance, statistics, and find the answer related to our research questions.

4.3.1 Gender

Gender
111 responses

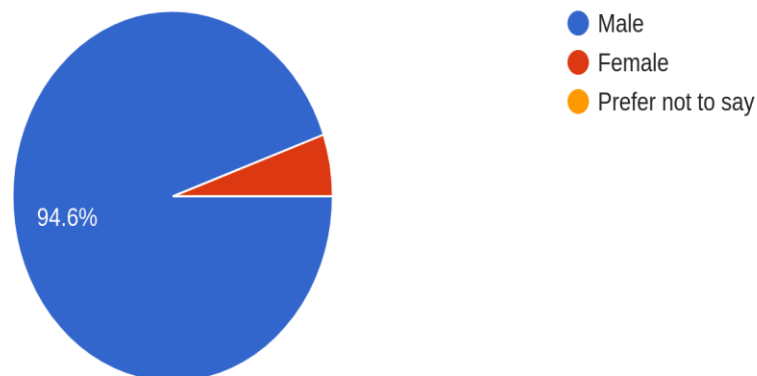


Figure 4.1 Gender of respondents

The above chart 4.1 indicates that the gender distribution was 167 (94.6%) male and (5.4%) female. This clearly indicates that most of the programmers are male and numbers of girls are a lot behind in programming.

4.3.2 Distribution of Age

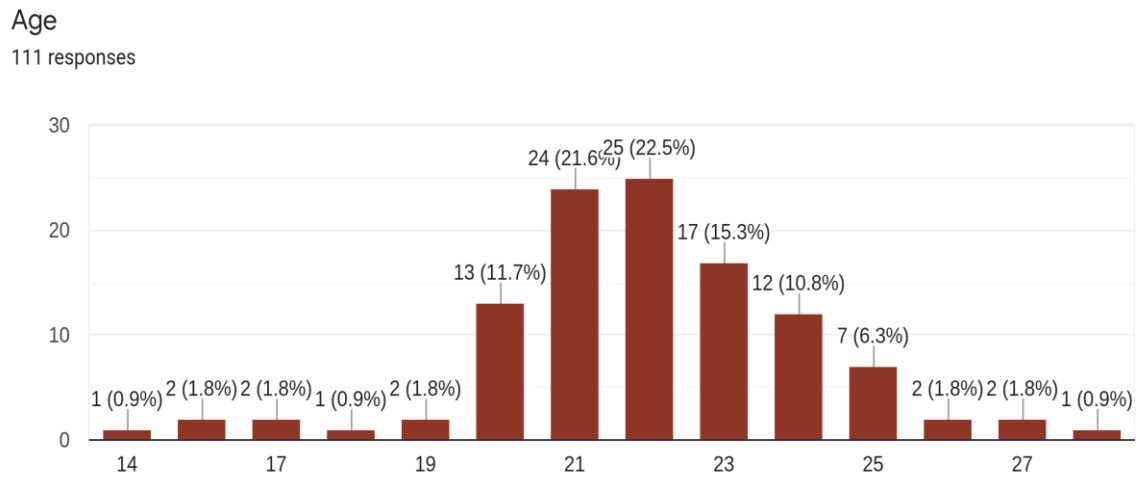


Figure 4.2 Age of respondents

What we can see is that most of the programmers in research in age between (20 - 24) in most cases are university students.

4.3.3 First hearing about Competitive Programming

When did you first hear about Competitive Programming?

111 responses

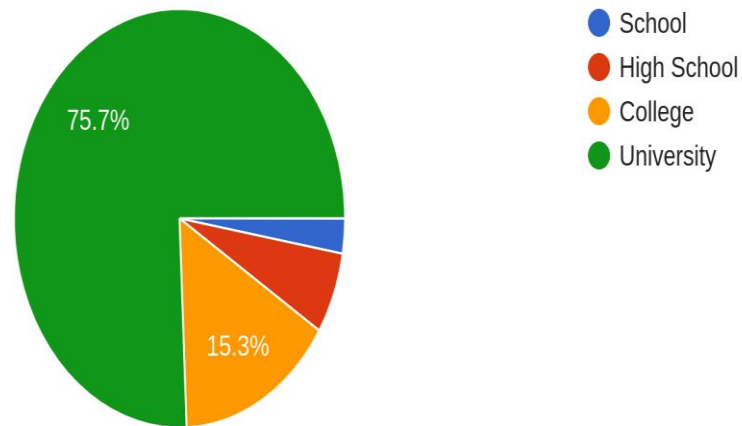


Figure 4.3 First hearing about competitive programming

From the above chart, we can see most of the Students hear about programming in university about (75.7%). If they hear from school or in college it would be a better time for them in the long term.

4.3.4 Competitive Programming Experience

Competitive Programming Experience

111 responses

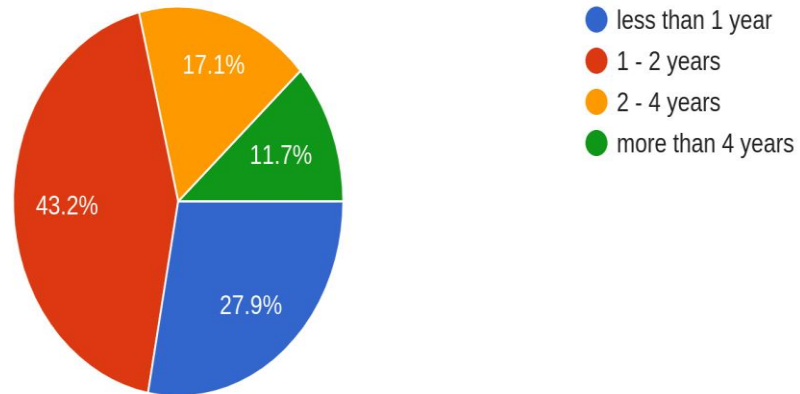


Figure 4.3 Competitive Programming Experience

From the above chart, they Can be divided into three major terms: Advanced, Intermediate, and Beginner. So we will get a lot of intermediates and beginners in the pipeline of the future. That's a good sign for our country's future.

4.3.5 Daily average Practice

Your daily average practice

111 responses

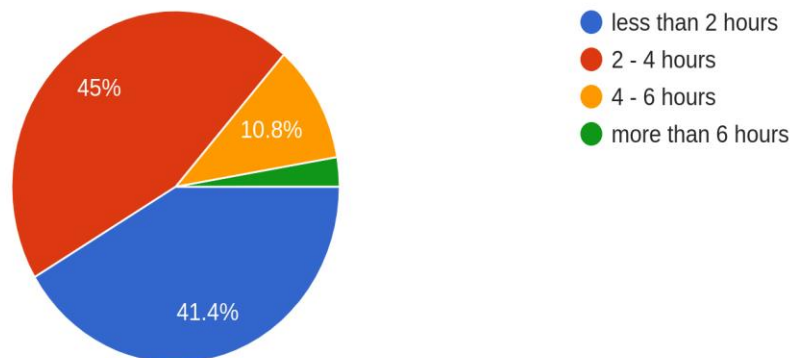
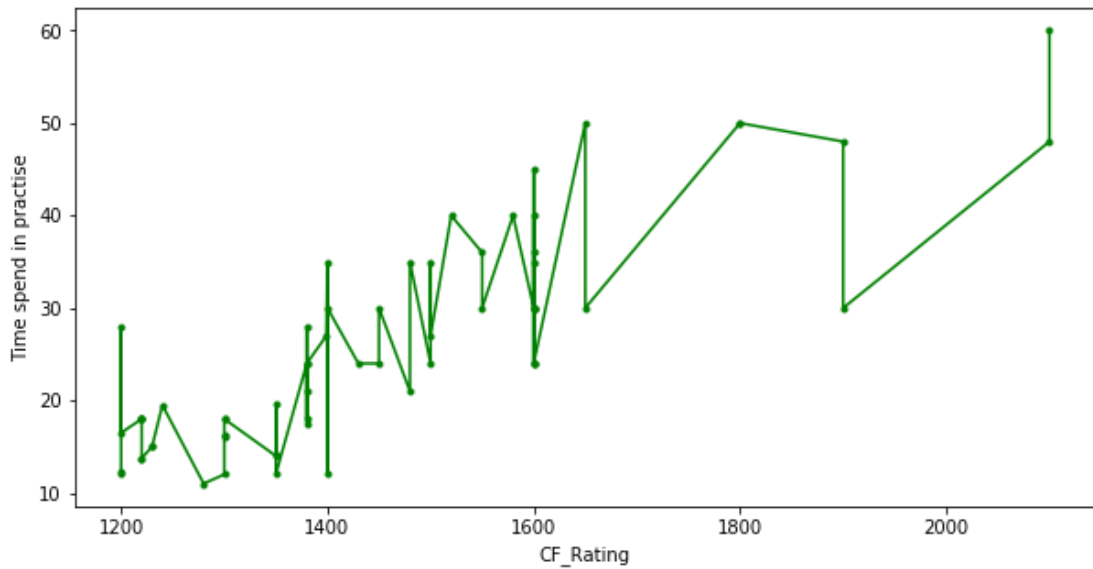


Figure 4.5.1 Daily average Practice

From the above chart, we can see that most of the programmers practice about (2-4) hours on a regular basis. And a majority of them practice less than two hours a day. But we will see next to the more we practice the better we get in programming and most of the intermediate and advanced programmers practice more than four hours a day.



In this graph we can see a huge difference between practicing more and less time.so in order to get the desired result we must do our hard work after all.

4.3.6 Time of Practice

When do you sit for practice?

111 responses

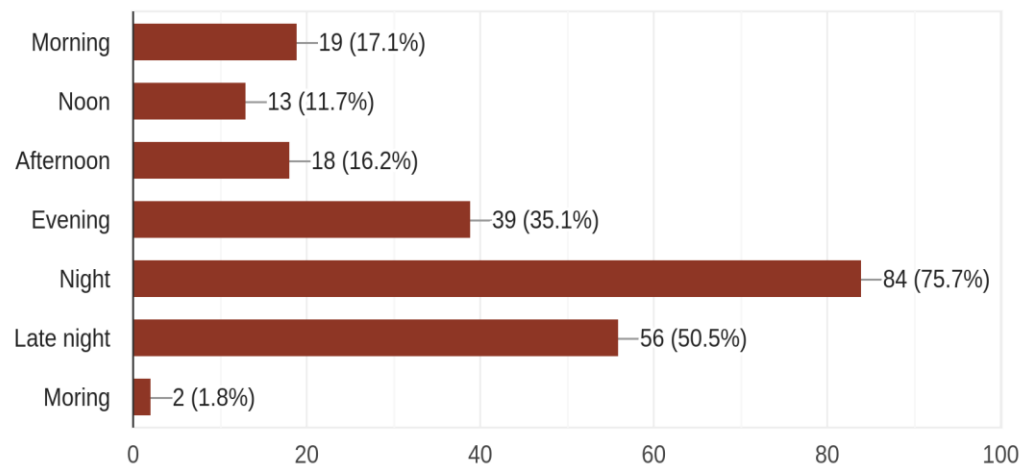


Figure 4.6 Time of practice

This study shows that most of the programmers sit for practice in night and late at night. The main reason for this is that all the students have to deal with their academics.

4.3.7 Learning sources

What are your top learning sources?

111 responses

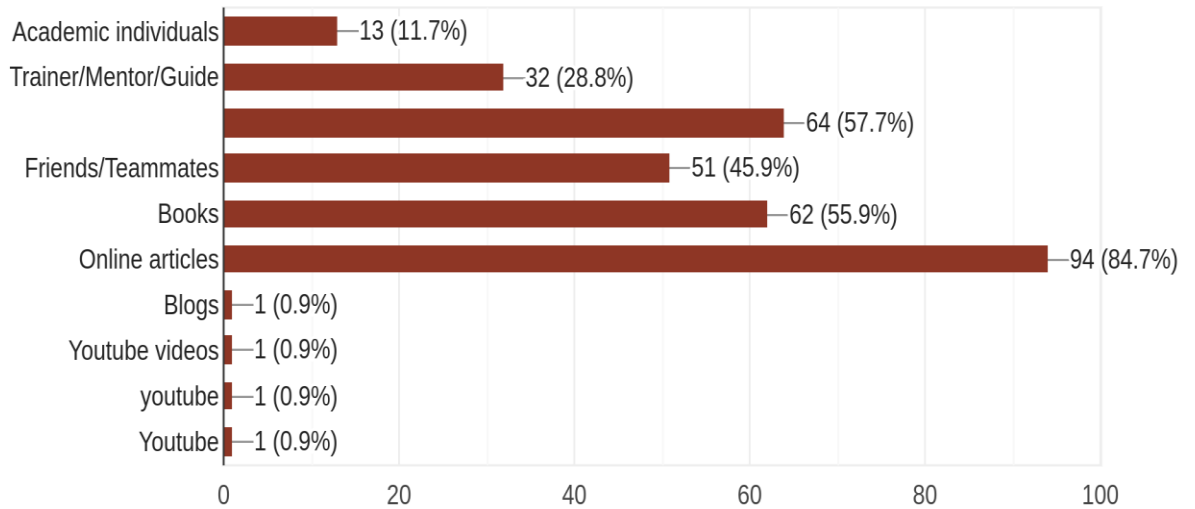


Figure 4.7.1 Learning Sources

One of the best findings of the Study is here. We can see that most of the programmers get the majority of help from online articles, Programming communities, Books, Friends and Teammates, Trainer/Mentor/Guide, And Academic Individuals.

This study shows that only 11.7% gets support from Academic Individuals and maximum Programmers who are now in a better Progress has the support of Trainer/Mentor/Guide.

I want to claim two things here: we need to increase our support in academic level and Practicing without having a Trainer/Mentor/Guide we got stuck in a sudden level.

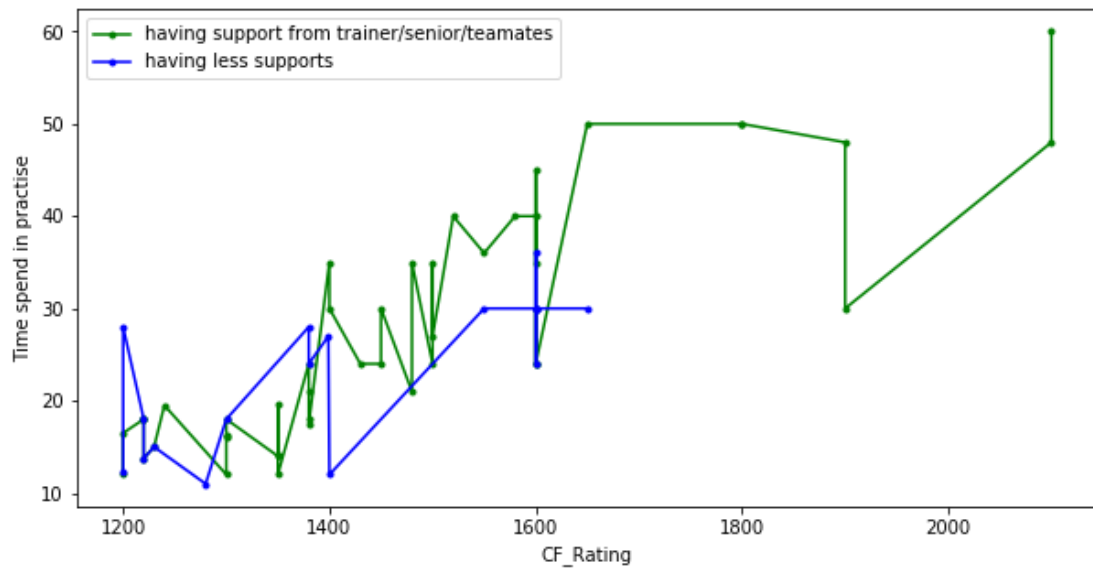


Figure 4.7.2 Difference of having trainer, seniors, teammates

In this case we can clearly see the difference of having a trainer, senior or teammates so that we can learn and spread our knowledge. In that case we can claim a thing here without guided practice we it's hard to progress.

Figure 4.9 CodeForces Ratings

Yet another sign of hard work and progress of programmers. The stats show that we had a lot of advanced, intermediate and beginners programmers, we should support them for becoming great programmers and achieving greats for our country.

4.3.10 Weaknesses in problem-solving

What are your weaknesses in problem-solving?

111 responses

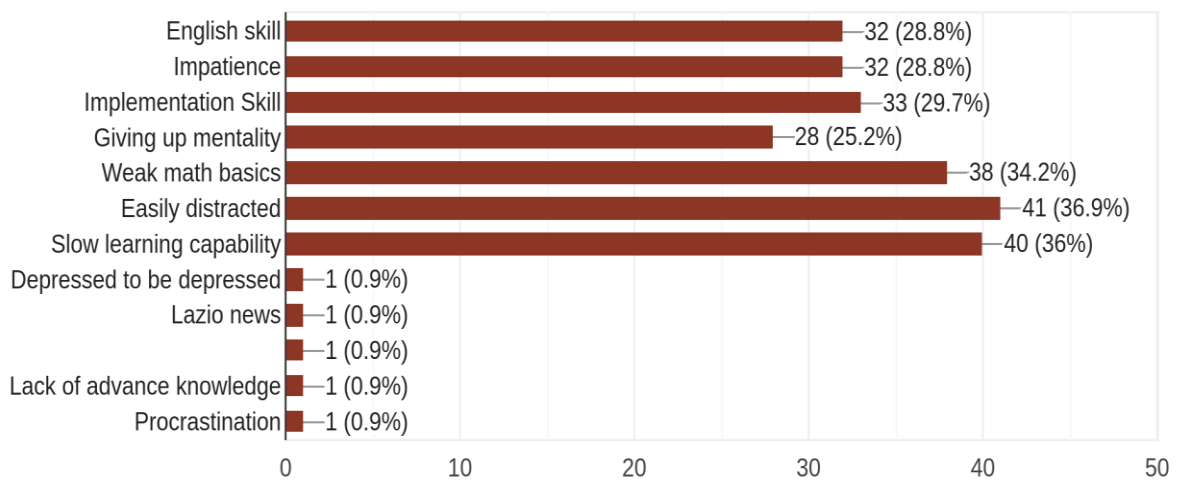


Figure 4.10.1 Weaknesses in Problem-solving

We can see here we have a lot of weakness to cover on. since we get strong at these things over time and hard workings but we also need to focus giving support to them to cope up on these things.

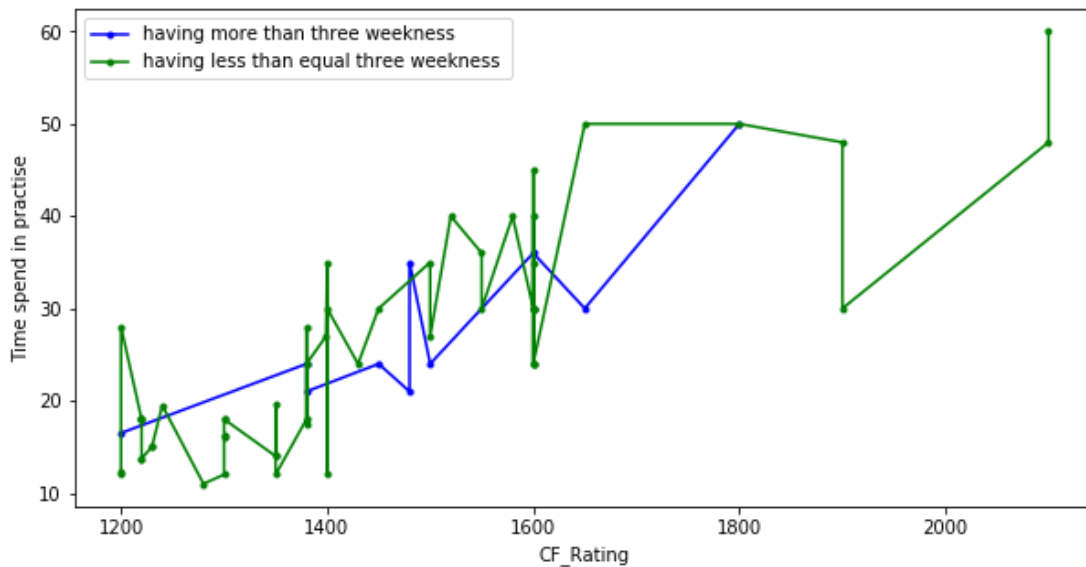


Figure 4.10.2: Effect of weakness over progress

So we can see here people have more weakness in particular areas and have made less progress over time and maybe they suffer and quit programming after sudden stages. We need to figure it out and give them necessary support.

4.3.11 What are your worst distractions

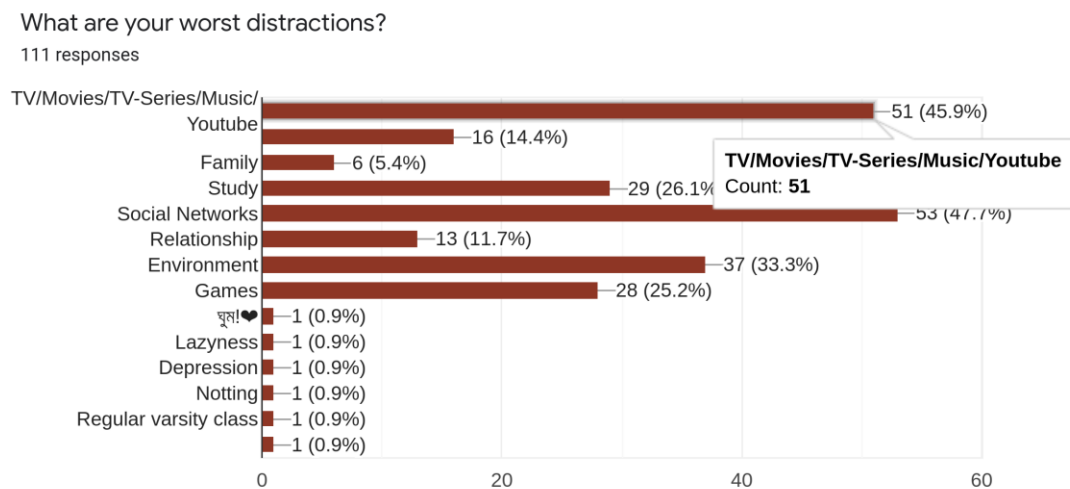


Figure 4.11.1 Distractions

Here we can see that most of the time we are distracted from online activities like multimedia, and social networks. Sometimes even from the environment and socializing.

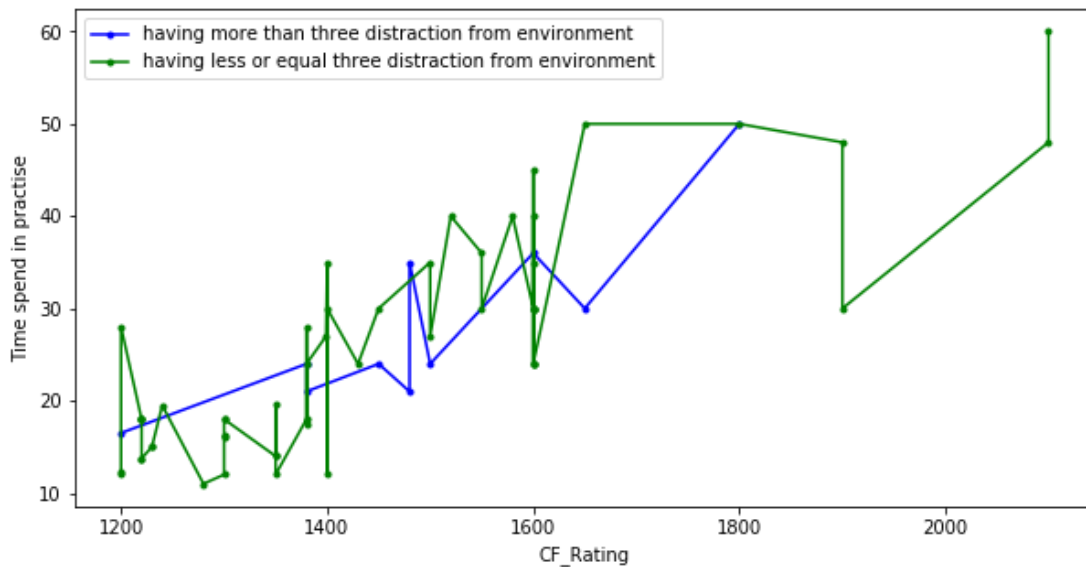


Figure 4.11.2 Effect of distraction over progress

So, we can see here people with more distractions from the environment lacking from their desired progress. Here we can also see that we are not able to give as much time as others having less distractions. So, in my opinion it's not a case they are excusing. We should give a lot of support to them to avoid distractions.

4.3.12 Reasons Behind doing competitive Programming

Why do you do Competitive Programming?

111 responses

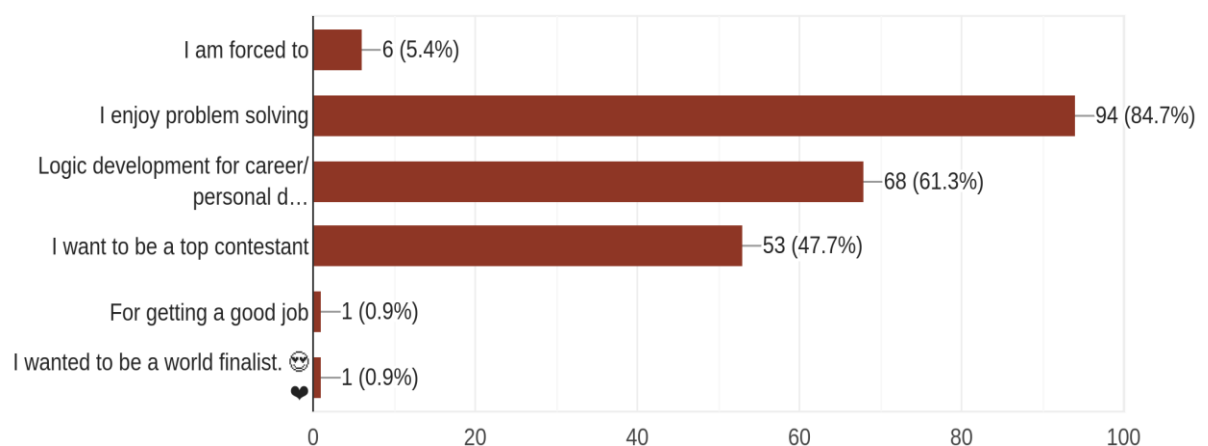


Figure 4.12.1 Reasons of doing Competitive Programming

It seems like people really enjoy problem solving. And their enjoyment leads to their goals and desires.

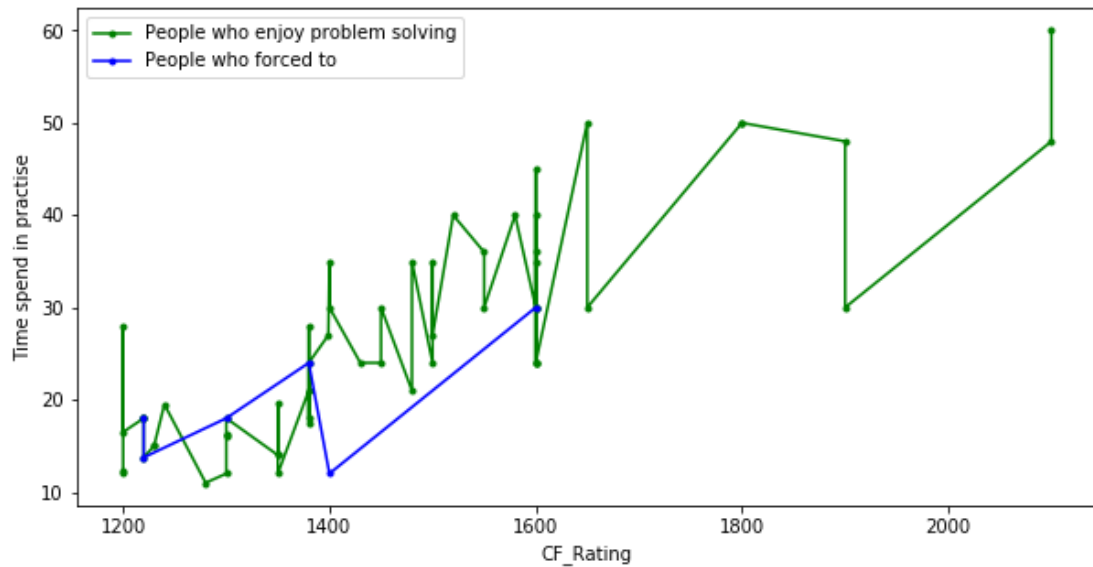


Figure 4.12.2 effect of mental happiness.

Yes it's more obvious to do things that we are loved to do. We better carry our own happiness. So that we can lead to our destination without having mental troubles.

4.3.13 Knowledge Sharing tendencies

Do you spend some of your time for teaching/helping/guiding others?
111 responses

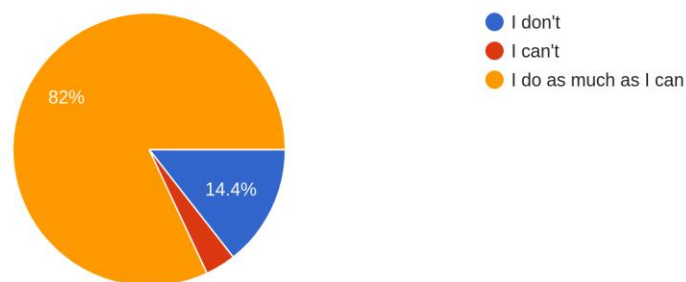


Figure 4.13 Sharing Tendencies

As we see in Previous research We have got huge knowledge tendencies among Programmers. In our case that stands (82%). I think that's more than any other communities.

4.3.14 Academic Study vs Problem-solving

How much do you like problem-solving compared to academic studies?

111 responses

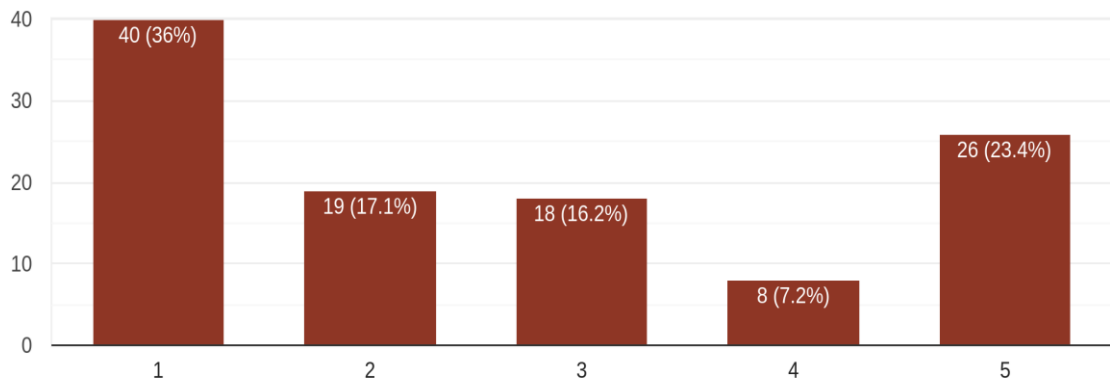


Figure 4.14 Academic Study vs Problem-solving

Yet another tendency among Problem solvers. They enjoy problem solving compared to academic studies. Yes, it's more challenging and enjoyable compared to academic study. That might be a reason for problem solvers to not focus on academic study.

4.3.15 Residence

Who do you stay with?

111 responses

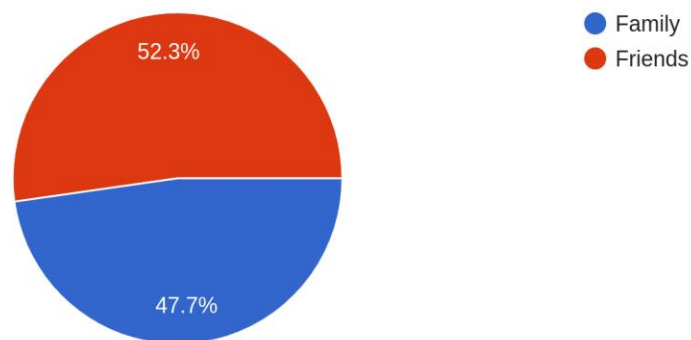


Figure 4.15 Residence

It looks like a tie with the residence and we want to look at the impacts of residence in further study.

4.3.16 Financial State

Financial State!!

111 responses

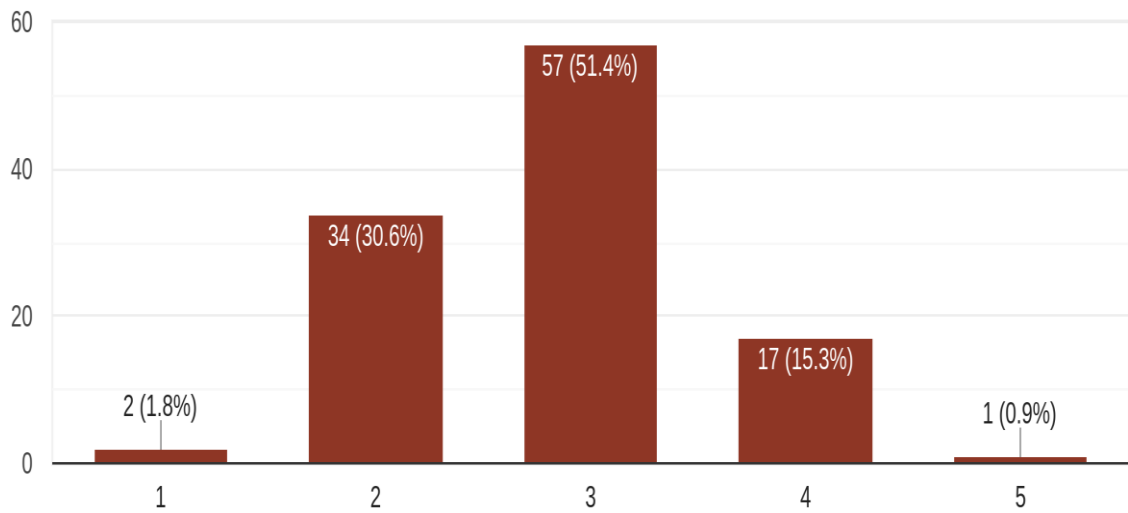


Figure 4.16 Financial State

It looks like we all belong to the middle-class family. A study mentioned in book Competitive programming 3 that students with middle-class families are more likely to do well in programming and in our research we find the same. If we look closely at the data they are doing so good.

4.3.17 Communication skill

How you rate your communication skill
111 responses

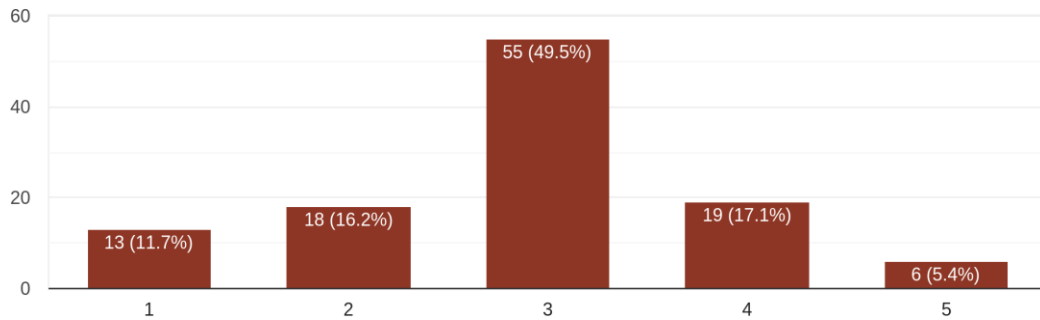


Figure 4.17 Communication skill

As we can see, the communication skills of programmers are average. We need to focus on this to build it up a lot more. That will bring a well benefit in the long future.

4.3.18 Relay on Competitive Programming

Do you think problem solving has helped you practically in some way?
111 responses

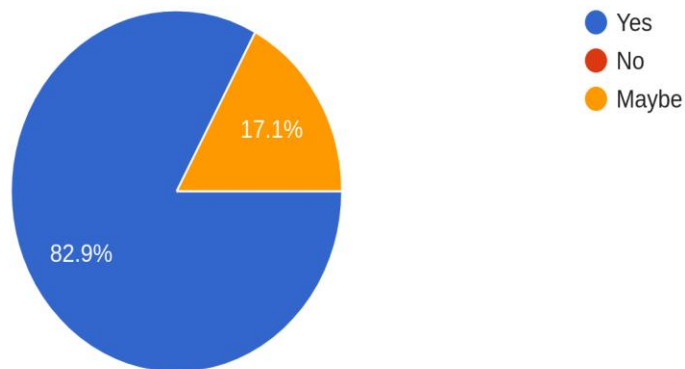


Figure 4.18 Relay on Competitive Programming

Definitely we should focus more on Problem Solving. A huge number, about (82.9%) says that it helps them practically. And nobody denies it.

4.4 Comparison

As the previous research deals a lot with the reasoning of programmers, their skill ups, and learning methodologies, their behavior, and knowledge sharing tendencies.

In our research, we work with a lot of domains the reasoning behind their doing well, their performing bad, effect of environment on them, their struggles behind their learnings, their need of necessary supports and aside we have also revalidate the reasoning of their programming, their skill ups and learnings, and their behavior and knowledge sharing tendencies.

4.5 Summary

In our research, we have tried to resolve and find the answer to some questions and try to uplift a community and show that they deserve support and the experimental results imply that we can find the answers, reasons of back and forth of programs and most clearly it showed that we need to support to do well in the future. And also clears the duty of individuals and authority.

According to the study, this is very satisfactory to find out some reasonings and causes of the back and forth of a community and their doings. This is a clear indication that the way we approached is doable and we've already reached a level of success.

As we all know, there is always a space for improvement, making more relationships, searching for more reasons we would require further for maximum output.

CHAPTER 5

SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

5.1 Summary of the Study

Through the whole study, we are trying to support a community of problem solvers. We are trying to find out the reason behind their back and forth. We think we have discovered the reasons behind their back and forth through the research.

We can see that they had gone through a lot of hard work and a lot of struggles over the years. We also see that due to lack of support they feel distracted, they feel hopeless, but they have carried a lot from their enjoyment over programming and problem-solving. Yes, it's voluntary work but it helps a lot in building up a career. We have seen that most of them say that problem solving helps them in practical life.

Finally, the community, every individual needs more support to give their best. We have also proven that our support and their hard work can lead us to different levels.

5.2 Conclusions

For us, it seems like a unique work. we think the scope of the actual research is very large, it may take years to perfect. Yes, it's a work that comes from the very deep of heart being a member of such a wonderful community.

We would love to say the finding of the study will help a lot to the community of problem solvers. If we can give our best support and they can give their best potential we can expect something really big going to happens.

I hope every individual and authority we are encouraged with the findings of the research and willing to accept to do their best.

Hopefully, someone new will carry the research into its next level.

5.3 Recommendation

We think no approach is 100% perfect. It has gone through a lot of research and development to overcome its limitations, constraint, and scope of improvement. Here we will see some limitations and future scope of the study.

5.3.1 Limitations

- First of all, we would like to acknowledge we tried our best to make an organized dataset. We were considering the time it would take to complete the survey to get valuable responses. It was hard to find in more detailed data to work on.
- Secondly, we found it's hard, clean, and processed the type of data we have found from the survey.
- It was hard to relate the data fields and find out the actual results. some of the cases we didn't find or can't validate properly due to the lack of a huge data set.
- We would be more graceful with the research if we can implement an organized model to organize, process, validate, and mostly interpret the data.
- We tried to cover a lot of things but there might be some uncovered stuff.

5.3.2 Future Works

- We could create a more balanced data set that can data from its core. Along with questionnaires we could arrange an in-person interview to collect data more precisely. It will take some effort but all the effort will be worth it.
- We could build more advanced and organized models to clean and process data. so working with big data is possible. And lead us to more precise results.
- Collect a lot of data from the root end, so that we can analyze a large programming community and find out what fits best for them.
- What we found most challenging through the research is finding the relationship between the data fields. It would be better to re-analyze and try to find some more interesting relationships among the data fields.

- We should also give a greater focus on the learning opportunities, ways of learning, and ensuring the maximum support for them.

5.4 Implications for Further Study

The research can be carried into its next level. we would like to see a lot of studies within this domain. some extra efforts in learning methodologies, improving weakness, removing distractions, coping up with emotions. Also, a well-developed methodology for data analysis.

For the supportive level a tracker system to ensure best support, measuring system for progress and comparison tools. Developing some methodologies for learning and preparing the best learning and sharing environment.

REFERENCES

- [1] E. Soloway, J. C. Spohrer, “Studying the Novice Programmer”, New York, 1989

- [2] Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, Pekka Abrahamsson, “What happens when software developers are (un)happy”, Journal of System and Software, volume 140, June 2018, pages 32 - 47

- [3] Ming-Ten Tsai, Nai-Chang Cheng, “Programmer perceptions of knowledge-sharing behavior under social cognitive theory”, Expert System with Application, volume 37 , issue 12 ,December 2010, Pages 8479-8485.

- [4] Ruven E. Brooks, “Studying programmer behavior experimentally: the problems of proper methodology”, Communications of the ACM, April 1980

- [5] Matthias M. Müller, “Do programmer pairs make different mistakes than solo programmers”, Journal of Systems and Software, volume 80, issue 9, September 2007, Pages 1460-1471

APPENDIX

Appendix A

Matplotlib

Matplotlib is a library for plotting and visualizing data from various types of numerals. It's an extension of numerical mathematics of Numpy. It provides object-oriented API for embedding plots into applications that use general-purpose GUI toolkits

Appendix B

Scikit learn

Scikit-learn is a machine learning library of python. It features classification, regression, and clustering algorithms. It includes support vector machines, random forests, gradient boosting, k-means, and DBSCAN. It's designed to work entirely with Python numerical and scientific libraries NumPy and SciPy.

Turnitin Originality Report

Processed on: 30-Jun-2020 19:04 +06

ID: 1351779328

Word Count: 5772

Submitted: 1

SP19D106 By Md. Mehedi Hasan
161-15-7068

Similarity Index

22%

Similarity by Source

Internet Sources: 18%

Publications: 7%

Student Papers: 18%

3% match (student papers from 31-Mar-2019)

[Submitted to Daffodil International University on 2019-03-31](#)

2% match (student papers from 05-Apr-2018)

[Submitted to Daffodil International University on 2018-04-05](#)

1% match (Internet from 31-May-2020)

<https://pt.scribd.com/document/258579958/Implementation-of-Lean-Production-Practices-in>

1% match (Internet from 20-Apr-2020)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3905/P14336%20%2814%29.pdf?isAllowed=y&sequence=1>

1% match (student papers from 24-Jan-2011)

[Submitted to Higher Education Commission Pakistan on 2011-01-24](#)

1% match (Internet from 07-Feb-2013)

<http://www.sciencedirect.com/science/article/pii/S0957417410004409>

1% match (Internet from 13-Jul-2019)

<https://ewic.bcs.org/content/ConWebDoc/3723>

1% match (Internet from 03-Aug-2019)

<https://pdfs.semanticscholar.org/4bab/b6cdf1f39a728bc325a4d930cd2cb6587fb4.pdf>