Introduction to Research

Prof. Prathap Haridoss
Prof. V. Kamakoti
Department of Metallurgical and Materials Engineering
Research in Computer Science & Engineering
Indian Institute of Technology, Madras

Lecture – 35
Introduction to Research

Prof. Prathap Haridoss: Hello, we are very happy to have with us today Prof. Kamakoti from the Department of Computer Science and Engineering, for this department module on Introduction to Research. He has been a faculty in the Department of Computer Science and Engineering for 15 years and has therefore you know interacted a lot with lot of students who have come through the department over the years and also has a lot of interaction with you know researchers around the world and the industry. His areas of research interest include software for VLSI, reconfigurable systems design, compute to architecture and secure systems engineering. So welcome to this class, discussion.

Prof. V. Kamakoti: Thank you.

Prof. Prathap Haridoss: So I would to begin with asking you, you know computer science and engineering has been around I mean for a while and lot of people at least in India, there is a lot of desire to be in computer science and engineering education process. So what are in terms of research though, in going away from under graduate education going into research? What are considered as traditional areas of research in computer science and engineering?

Prof. V. Kamakoti: So Computer science and engineering as you rightly told is a most desired field of many, many people in the country. Reasons are many, if you look at the way man lives today, the actual intervention of computing devices is increasing in a rapid pace. One of the main areas; let us start with the one of the main area which internet of things. Today the entire health, the entire schedule, everything that a man does could be actually monitored, recorded for basically providing some sort of service to the person or to actually for other monitoring purposes using devices.

So, the electronics and basically the compute part of the electronics has become closer to the man kind and that has opened up several you know research areas. The traditional areas if you look they are hardware, software applications and theoretical computer science. In the hardware, basically people were looking at how to make hardware that will work, that can be sold in the market and specifically they were looking at how much power it will consume? How much weight it will be? And how much heat it will be dissipate? To basically see that it is deployed in some environment, these are traditional areas of research when you look at hardware.

The second on the software side, people are looking at how to build, how to actually provide certain infrastructure by which people can develop you know software on the system. Software is of two types, one is the system software which basically comprises the operating system, which comprises the compiler and then you know certain build environments like, for example, certain design kits that we call and lot of a research has gone there which we called traditionally a software engineering of how to make tools that can enable people build large scale software.

The other aspect of software is the applications, where you actually build applications that suits different environments and governing all these three is the theory, which is the foundation. The theoretical computer science basically tells you that, if you want to solve such a problem what is it that you can expect, how much time it will take for a system to execute this particular, to solve this particular problem. So, you get some ideas and based on that ideas you refine your solutions and before actual you start implementing a solution the theory basically gives you lot of insight into how well your final solution could be. So to answer your question in to some of with answer for your question, there are four major areas of research traditionally and it continuous today hardware, software, applications and theory.

Prof. Prathap Haridoss: Ok right, I think nicely summarized in that way about what computer science, in general is in a traditional sense. In the same context, I mean of course, even in traditional areas of computer science like in other form I mean disciplines in engineering. Even in traditional areas I am sure there is you know the latest you know versions of what is happening, and how people work on it. But in addition to that, in addition to all these areas and the you know cutting edge of those areas, are other areas of research in computer science that are considered new recent areas of research?

Prof. V. Kamakoti: Yeah so, as I started just conversation one of the new areas of researches is internet of things. Internet of things, basically the research there is that I need to monitor something; I need to sense them data for whatever purpose it may be. For example, I need to monitor traffic, I need to monitor pollution level, I need to monitor temperature, I need to monitor say tsunami today. Now, how do I sense that data, how do I take it out and then send it to a global monitoring place, a communicate over the network and then somebody goes and use that data and say there can be a tsunami, the pollution level is high. So there should be somebody who makes a decision based on the data so this entire notion of internet of things as opened up several challenges in the traditional area of research.

For example, sensing is a interdisciplinary topic so we start now interacting with say people who make sensors. Suppose, I am monitoring the structural health of a building, I start interacting with a civil engineer okay, when I want to you have a car which as smart hybrid vehicle system, you start interacting with IC engine person to basically understand. All these are data collection points and that is in now becoming an interdisciplinary research of how I effectively and efficiently and accurately I capture the data. Once the data is taken, then I need to have mechanism by which I do some local processing , many times, so that comes under the hardware design, one of the traditional areas and that challenge is there is that I need to process that data with limited amount of competent. I cannot put a server at every traffic point, suppose I am monitoring the data.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: Monitoring pollution, I cannot put a server in every signal. So, I need to have some very small devices, which can work probably in a solar type of environment and I need to do some processing, then after that I need to send that data to a server somewhere, that is a problem in computer communication so that is a big problem in communication. Because I may have limited band width and may not even have network connection, say suppose I am doing a remote sensing in high altitude area.

I may not even have network connectivity so how I gather the data and bring it, so that is another challenge that is coming.

Now, IOT is one very, very important challenge, now once you get this data what you do with this is data that becomes a problem of plenty. Now I have to handle this data, I need to visualize it, I need to processes it quickly. For example, if there is a monitoring traffic, I need to tell very accurately within the next of few seconds that there is a traffic jam here. If I say 1 hour later, it doesn't make sense so there is some real time requirement for me to quickly accumulate that data, get to some conclusion about data, make a decision and communicate that decision, make an observation and communicate that observation out. This opens up a vast field called data sciences right or data analytics so that is another field, data sensors, data analytics becomes one very important field.

When you look at that field, now there is software by which I need to process the data, I need to write applications to actually extract the data and then this cannot be that just by viewing say millions of bytes of data, for me to come to conclusion is very difficult. I may have to monitor several situations and then come out with some observation and that monitoring mechanism comes under the purview of what you call as machine learning right. So there are machine learning engines, who would have studied this patterns for a period of time and then they will now say this pattern now is coming so this is should be matching on to one of those earlier things, there is a Training phase, machine learning is a another thing that crops up because of this internet of things.

Then covering all these things is now what we call as the information security, for example, monitoring a pace maker of an individual is now done by a device which can be remotely operated. So, it takes your heart parameters and it sends over a network and doctor can monitor. Now, if somebody could penetrate in to this network, a hacker can penetrate in to this network, now you can just go and basically stop the pace maker essentially you can do a cyber murder today.

Prof. Prathap Haridoss: Oh Okay.

Prof. V. Kamakoti:. So that opens, as and more your electronics become close to your heart or close to your body and it becomes more penetrations, its penetration into your basic living becomes more and more increasing security becomes a major concern. Now, security is a habit, if you take human being, if that habit is not inculcated at the age of 5, you cannot inculcate it till symmetry, this is saying in Tamil [FL].

“ Anjula Valyadhadhu Ambadhula Valayadhu “

“ Thottil Palakkam Sudugaadu Varay “

Prof. Prathap Haridoss: Yeah, yeah.

Prof. V. Kamakoti: So these are all some statements we will see in Tamil, so the habit that you inculcate in the age of 5, will be thing. But, now computers are in its say around half a century, software development started, big software development started late 1960s even before we both were born, in our previous batch so that is at least now 50 to 60 years old today. Till now people have not been looking at security in a big way and suddenly the security which is like a habit has to be retrofit into this over grown old man.

Prof. Prathap Haridoss: Ok. Yeah, yeah.

Prof. V. Kamakoti: And that becomes a big problem. Today, if I ask you if this mobile phone secure? I do not have an answer. Will my data in this mobile phone be stolen or if they ask you a question, can somebody hack and get some personal details from this mobile phone? I don’t have an answer right because this system has so much over grown and I don’t know how to basically go and certify that it secure. This information security suddenly with boom of internet and suddenly people realizing that internet can be used for malicious reasons also, information security is growing up as one of the major areas of research today. So whatever you talk this is going to be one information security should be there so everything finally boils down to how secure is the information.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: Okay the next thing that has happened or a period of time, so we have covered something on application and something on a software and machine learning and all these ends. The other important thing that has come up specifically in large scale computing, what you called as high performance computing is that, we could not increase the frequency of the processor, clock frequency of the processor. As you see we are not seeing any processor that is running more than say 3 to 3.5 giga hertz, the reason is that the power is directly proportional to the frequency and so if I keep increasing the frequency, the power will be so high that it will burn the processor itself, literally burn the processor itself.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: People have now moved on from a single core model, a single CPU model to multiple CPU so I cannot run give you 10 giga hertz. But I can give you 4 CPU each with 2.5 giga hertz right, so that is possible for me. Now, you cannot actually find a processor CPU today with just 1 core, it is all multi core. The moment I have multi core, then one of the important thing that we need to research upon is, how do I make all the 4 cores work together?

So if your program is a sequential program, namely first step gets executed. So suppose, I want to add 100 numbers, suppose I say add the first number to the second number then to the third number then to the fourth number, this is a sequential program. Now, I need to say take 4 chunks of 25 numbers each, make the first processor add 25, second processor add 25, third processor add 25 and then collect a result and give one answer so is the difference between sequential program and a parallel program. Now, parallelization is now became another major area of interest right so we have to take a core program and see how I can make it into concurrent pieces which can execute at the same point of time and give a result. In nutshell, these are all some of the you know research, new research areas, these are very nascent in stage and there lot of gold mine of open problem for people.

Prof. Prathap Haridoss: Ok, very nice theory, very thorough perspective of; very detailed perspective of what computer science is today, computer science and engineering. Maybe in this relation, since you both mentioned traditional areas and you know modern areas that are coming up in computer science. If you look at the industry, see they I would like to look at this from say the perspective of the research students who finish an MS or a PhD degree, what sort of activities that go on in computer science are of sort of more immediate interest for the industry, where you know students do research it's also something that they can see immediate applications to?

Prof. V. Kamakoti: So industry today can be broadly classified again in based on the areas of research. For example, traditional hardware design, where there is a lot of stress on today on how to make hardware that is despite very less power, how to make architecture that can suit different environments today. So, one which can do, today if you take a network appliance which for example take some packets and route it, it is also a computer. How do I make device that are very small and that used in millions for the internet of things type of application. There is a set of people, set of industry which are looking at these types of hardware designs of low power, high performance hardware and they are looking at different models of computation.

Today there are notions what you call as approximate computing, stochastic computing, non bunion computing, so different computation models are there and the research labs within these core industry have started looking deeply into this, so this is one line of that. Then the second line is that the systems are becoming very complex, so how do I model a complex system and then how do I verify it is working correctly. There is, how do I make a hardware which lots of complexity so the CPU that was thought for entire semester, in my B. Tech curriculum in 1985, today I teach the same thing within the first week.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: There is bigger challenge here because history cannot be forgotten and at the same time.

Prof. Prathap Haridoss: Progress is.

Prof. V. Kamakoti: Progress is so rapid that what I learnt for entire semester, I need to teach it within 1 week here. So, essentially the systems are becoming more and more complex and how to model such systems, how to verify such systems, so that is another area. Hardware if you take, there is hard core design and hard core modeling and these two have lot of open problems and lot of job opportunities, so this one set. Now, when we come to the system side, that is the operating system, there are very few groups which work on operating system, there few very individuals in the country who can actually going to a kernel and come out with modifications to the kernel.

There are very few across the globe who can do, few in the sense in terms of percentage it will not even cost 1 percent of the total computer science engineering people who would be graduated, so because it is very, very complex. If people start working on these type issues like kernel level and the compiler basically optimizations and lot of system programming stuff, there is a lot of potential for them in terms of going into a very core research jobs be it a faculty position or an industry position or even a government research establishment versions, so that is also very, very interesting problem in terms of research, in terms of publication, in terms of livelihood everything it's a very good thing.

The next thing is applications, today application wise if you look there are two ends, one is an elephant namely data sciences. Data sciences today is formed between anywhere from, if we have 16 departments in our institution, every department has a data science related problem and that is proved that we have data science laboratory today which has 22 or 27 faculty members across all the 16 departments we have today, that is massive field that is one end.

Another end of this spectrum is, we have this IOT type of devices; in the IOT know the memory will be very small. I cannot write large scale programs, I need to have very effective coding efficient, coding techniques and stuff like that so writing those types of small programs on the devices that's also a very big today, that's a big skill and that we need look into it, so that is from the application side. If you look at hardware, if you look at system software, you look at applications, these are all the big research areas and of course, on the system application side, writing these type of parallel programs also is a skill and very, very few people across the globe has this type of exposure. The very few (

universities even offers such type of parallel architecture and parallel algorithm courses right and so these are something that great research interest here and if somebody aspects to do that, if they take anyone this area and spends 2 to 3 years and demonstrate to the world that they can think new, publish couple of papers write some sort of software I think a they have a great future in front of this.

Prof. Prathap Haridoss: That's I think very good perspective for you know incoming students to consider. Now, over 15 years we have seen several students come in and graduate not may be directly in your group and also around in various groups so on. In your opinion, see there are already some metrics that people use for gazing how you know progress in search is going on and how successful students are in research. In your view what are the parameters that you would look at to a two gaze a success in research that may be student should also consider looking at right?

Prof. V. Kamakoti: If you had asked this question, if you had asked this question in 2001 in 2005, 2010, 2015, my answers have would be different.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: 2015 the answer which or probably 2020, the answer is that today see those days I could very well say I am a hardware engineer and then forget about everything else and things worked for me, like I could have a career based on hardware engineering just hardware I could go to company which look only a hardware and I could have a career successful career, but that career would have been short stint of say 6 to 7 years, now after that thinks start a things. Today I cannot say I am a hardware engineer everything today is cross layer.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: So if you look at traditional computer science it should be therefore, any device you take in computing I have something called digital circuits and on top of it I am talking about a digital computer I have a digital circuit there is a micro architecture then an operating system then system software then applications. So, this is tag for me to even go and tweak some parameter, for example, temperature control at it circuit level I need to know the entire stack I should write a thermal aware compiler, I should have a thermal aware operating system, I need to have thermal aware micro architecture, I need to have a thermal aware digital circuit. So, anything that I want to do anywhere in computing I need to have the entire knowledge of the stack and then only you get, you become a full person and if you don't have that knowledge then it sometimes it becomes a quite you know.

Prof. Prathap Haridoss: Shaky

Prof. V. Kamakoti: Shaky you can be bold over at some point of time. So, my sense of success today is if a student is able to do a research work where he demonstrates that whatever he has found out in whichever layer that 5 layers I told right digital circuit, micro architecture, operating system, system software application, each of there you would have some enhancement, but he should know demonstrate that is well there is a synergy of this and it will effective in all the 5 layers and he is able to argue it and bring it out, I think that is success with respect to and that is also the challenge right.

So, today is the challenge essentially comes normally I give questions in comprehensive viva and typically end sems to design as, they will give a very good answer, but it will take 2 billion dollars to implement. So, that is not engineering. I need to make a hardware or I need to make a system that could be effectively implemented that effectiveness is just not performance, but it is also cost and sell ability in the market. So, might have a very good architecture, but it cost that machine cost 1 billion dollar probably 1 fellow will buy it.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: So, then it will not be successful. So, to have that a notion of cost performance and have the notion of you know the entire stack and it should be feasibility that is very, very important and that is how we, I feel that a particular research is successful only if it addresses, at least in computer science it addresses this spectrum.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: Yeah.

Prof. Prathap Haridoss: And now I mean in that context in terms of the context of say successful being successful as a researcher and so on, when I think in general there is always an expectation there is always this feeling that you know when somebody does a MS or a PhD degree they are now specialist in that area and maybe that restricts what kind positions they can go for and that is always a concern that students have when they pick up in an advance degree in any field. In computer science what sort of positions do students who finish MS and PhD degrees tend to get?

Prof. V. Kamakoti: Okay So let me just take, I did PhD in absolute theory I have not done anything on systems but then, I got a job I worked for two postdocs they were more or less theory but then I got a job in the industry which was on processor verification is completely hardware then I came back and I did VLSI here and the today I also consult for a financial bank and I am able to do some you know IT activity there. So if you take essentially we started of the theory, but we are able to do hardware we are able to do application.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: Just looking at myself that my choice of talking theory that time did not actually restrict me explore all these opportunities.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: Whether I am successful or not we are some more years to tell that but now, but today I don’t feel some handicap because I took theory there so that is reason why I put myself as a case example here.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: I think computer science is very broad, actually I have seen very good software engineers even from our own institute they are chemical graduates.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: And interestingly three of big shots I know in that field of software who have been vice president of multi software that goes on to amazon, why an amazon or yahoo or this type of sites, right if you put there within the next few seconds there will be one million hits or something. Whether you are software is working or not will be known within the next few seconds by time you wink your eyes three times your software stability will be not. So these types of projects people are handled and all of them are non computer science people. If non computer scientist could survive in a bigger market, I think somebody was reasonable introduction to computer science irrespective of what they have done databases or whatever they have done I think will certainly a have a very good future there. So at least from computer science point of you people need not really worry what area like it.

Prof. Prathap Haridoss: What area like, Ok.

Prof. V. Kamakoti: But there are certain core courses like data structures, algorithms, programming languages, computer architecture, operating systems these core courses are there basically the language of computer science, right. Basically automated theory these courses people either in the undergraduate level or in the post graduate level there should thorough it. What I mean thorough is there are some very standard text books for each one of them and NPTEL has lot of lectures people have to access it and there are very good problems.

I don’t know about other fields but one thing very interesting about computer science is, all these books have very interesting problems at the end exercise problems and people have to work out this exercise problems whether you are a teacher or a student, it is a must that you workout all these a exercise work before you start teaching or you start researching. If those things are done then that foundation will be strong and then you can explore anything. So, that worry that if I take data sciences I may not able to do anything else is all unfound.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: Which some if you have this foundation anything you can learn and develop within a few years.

Prof. Prathap Haridoss: Great this about the people who we just spoke about the people who are just leaving the department with the post graduate degree. Now, I will step back a bit and look at the people who come in to a department for Masters or a PhD degree, in general do you see then facing any specific kind of difficulty for them to adjust to a research in computer science and engineering and if so, now what are your suggestions or how they are supposed to handle?

Prof. V. Kamakoti: Yeah. My suggestion to all people who are entering computer science department in general which I been coordinating the same research methodology course quite some time before and I insist during the department modules and I tell now, there is one theory like if you don't know the language in which you can communicate you can think, you cannot even think. So, even if I start thinking I think in some language. So if I start if I want to think about computer science concepts, the language of computer science should be there in your mind then you can think anything about computing.

Now, what is this language of computing you just basically formalism, right? Suppose the problem that I see even with students who are about to graduate is that if I ask them to describe something from LKG to plus two. They start with the LKG entrance, then plus two final exam, then some eighth standard half yearly and then third standard quarterly. The organization of ideas is not in a; it's very, very zigzag. What it means to organize that area, suppose I ask you to write a research paper or present something informally the idea should get organized and the way you can do is to basically learn the language of computer science and I strongly suggest that there is a course called Discrete Mathematics. Discrete Mathematics I think there are two offerings in NPTEL, one offering is by Prof. Kamala Keerthivasan and she has done it so nicely that she will write line by line and repeat every line twice and that type of course and she also has a book, her own book which you can use it and does excellent problems and there are many books in discrete mathematics also.

Take one book, good book and listen these type of NPTEL lecture and get yourself very thorough with Discrete Mathematics there is another book by Knuth, who is the father of computer science and he has written that's called Concrete Mathematics. These type of Discrete Mathematics and Concrete Mathematics if people look in to it just solve at least 20 percent of the problems in some of the standard book there that will give them the organization of ideas. So I want to write a proof. How should I go about? What sort of proof technique should I use? How do I go you know step wise; all these things will come. The moment you have that clarity of thought then I think it becomes very easy see which ever.

So, this is one problem which people face and somehow when we all did our PhD program, we use to listen to our guides. So, my guide was a Prof. Pandurangan he said do all these things we did all that 300, 400 problems in standard book or something like that and we all create. Now, that type of dictate is not working out with our modern students. So one of thing though we do not want to take parental role or a dictator role and say, do it. We will now request them to go and look at these two books and study this; this language of computer science has to come.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: If it comes in then any subject will become easy because you know how to think, then you can think about anything nothing is rocket science, this is one fundamental problem. Communication skill in this conversation, I would have made say some hundreds of grammatical errors with that mainly we are able to convey hopefully right? So this type of a real a Shakespeare English and all we do not need. We just need them to write the proof clearly, write the concepts clearly, and explain the concept clearly in the language of computer science. These types of training in Discrete Mathematics and Concrete Mathematics would be very helpful. NPTEL has made a great step towards Discrete Mathematics.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: And Concrete mathematics by Donald Knuth book we should follow this is my general prescription for anybody who wants to come to Computer Science.

Prof. Prathap Haridoss: Great, great, great that's very nice. The in any other engineering discipline I mean especially there is a lot of experimental work. I think the lot of people talk about you know there is the atmosphere in the group and how it impacts the research student and therefore, the meetings that students has with his other group members and the meetings that he or she has with the guide is a very important aspect. Along similar lines in computer science engineering you may be a bit of a mundane question, but how often do you think student should be meeting their guides and you know what sort of an interaction should be there?

Prof. V. Kamakoti: More than meeting the guides, meeting the peer group is very, very important.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: And then having a year for everything is also very important there should be very keen on observing, for example, I did a course on Distributed Computing with one Proff Mutthu Krishna of our institute, he retired now. But the first quiz, I could not even understand what he was telling when I went and asked the sir, I do not know how to handle your subject though I got reasonable marks but I am not too happy with this thing. Anyway go to Adyar signal and see how their traffic is thick talking about distributor, go to Adyar signal and see how traffic is moving here and there right observe it for 2, 3 days then you will understands more about distributor computing is true.

Actually it was true right. So, how coordination happens, how distributed coordination happens, how clocking, how signals work, what happens if the signal is not obeyed, what happens if there is a delay, what happens signals while so, many things talk about many, many coordination problems in the real Distributor Computer Science. So the observation also is very, very important. So I still believe now, So I have a team of around 40 students from B. Tech, M. Tech, MS, PhD put together and then we have 5 faculty, 4 faculty in our lab each one have something.

We have around to 100 to 120 people in the lab and that is very, very good. Almost two students review their paper each other and make our life very simple and we also now I have to scaled because we cannot say when we did B. Tech it was 20 to 25 in a class today we has 3 times there are thing or M. Tech programs have grown, our research areas have grown and there is so many projects also. As a faculty the amount of time we can spend on a student as also decreased but how do we compensate for that is to have a larger team and the team members interact and when, when something comes up to you it is after lot of refinement there and I have seen that.

So, when today I need not micro manage M. Tech person, if he is willing to work because the PhD student will actually micro manage him to a large level and no I have to macro manage him. So peer group is very important, going across boundaries and the basically talking to the next team understanding what is happening there computer science is all about applications. Today we exist because you guys have given us problems you guys are solving problems on me if I don’t have a user computer science would have gone it is a user grown field.

So people have to start looking at those type of things and that is very good and specifically in our institute with this interdisciplinary stuff that has come actually 4 or 5 interdisciplinary students are there in our lab today and that is creating lot of more you know existing problems. So why I have mentioning here is that you talking to your peers is much more important than talking to your guide. But at least once in a fort-night meeting the guide and also sending some daily reports or weekly reports saying this is what I have done even if you are not done anything I did’nt do anything.

Prof. Prathap Haridoss: Ok.

Prof. V. Kamakoti: So, that is where that will keep (Refer Time: 36:36).

Prof. Prathap Haridoss:. So, I would like conclude this discussion with you know, what are your words of advice for anybody who is aspiring to do research, pick up a research degree in you know computer science get into a research degree in computer science?

Prof. V. Kamakoti: My advice is that we need to work hard. Specific advice that is that is for any general.

Prof. Prathap Haridoss: Yes.

Prof. V. Kamakoti: Today as a professor my working schedule is close to 16 hours a day. So, the student I expect him to work double that time. So, go to Sun God and say extend 6 hours and make it 8 hours and make it 32 hours a day. Now, work as hard as possible that is very important today because the competition is very high. If you look at universities across, if you the problems if you are working on problems there are at least today 10 groups which are working on this same problem and this is survival of the fittest and they all send to the top tire conference and that the fittest one goes in and if you miss one dead line there will be three fellows at least who would have published in that area and your one year problem is gone.

So, that race is there always and you to be prepared to be work very hard and to establish yourself and then the next thing is team work is also very, very important that is that should from day one, work in a team to share ideas and be ethical about it because today if you look at many of the major universities across single author paper or student and that author is all gone today. If you take some really seminal papers there are at least 4 to 5 authors right, these all comes out of some large scale collaboration those are exciting area team work is very important and then my traditional advice of getting your formal theory correct your concrete and discrete mathematics very clear that is also very, very important computer science is a rapidly changing field. So, if you think of a problem today within 3 years you do not solve it becomes obsolete at a time you submit your thesis. We have to be on your tradition even I am not too comfortable guiding a PhD beyond 3 years because that original problem, you could have taken they have already become obsolete by 3 years. So, there is big change in field and lot of things especially in computer science is with the industry. So if I get a tool that I am working, this tool is 5 years old they would have tested for 2, 3 years and then release the alpha version to you and so that tool is already 5 years old. So, to keep in trend with the pace of the industry and you know making it irrelevant needs lot of hard work. And so finally, it's all hard work plus team spirit plus good foundation.

Prof. Prathap Haridoss: Thank you Prof. Kamakoti.

Prof. V. Kamakoti: I think this is interesting conversation.

Prof. Prathap Haridoss: Thank you.

Prof. V. Kamakoti: And thank you for selecting me.

Prof. Prathap Haridoss: And not at all. In fact