



CSI Connect

A tri annual newsletter of
CSI Student Branch
Aishwarya Institute of Management & IT
Udaipur



Issue : 1

January 2012 - April 2012

On 05 December 2011, CSI Student Branch was inaugurated by Shri Satish Babu (President CSI elect-2012) from Trivandrum, Kerela at Aishwarya College, Udaipur. Dr Azimuddin Khan (President, CSI Udaipur Chapter) updated the students about the objectives, benefits, working and upcoming events undertaken for students by CSI.

The technical talk delivered by Shri Satish Babu on FREE and OPEN SOURCE SOFTWARE (FOSS) and MOBILE COMPUTING, covered the basic nature of the software, evolution and advantages of FOSS. He also threw light on the significance of mobile computing and trends of Applications in the field were discussed. Udaipur CSI Chapter Board of Members marked their presence in the inaugural function. The students and faculty members' enthusiastically participated in the programme.



Delivering technical Talk



Audience attending to the talk on FREE and OPEN SOURCE SOFTWARE (FOSS) and MOBILE COMPUTING



Shri Satish Babu (President CSI elect-2012)
Inaugurating the Chapter

VIRTUALIZATION & CLOUD COMPUTING

Udaipur (Rajasthan), Tuesday 27 February 2012 - A Seminar on "Virtualization & Cloud Computing" was organized by Aishwarya CSI Student Branch, Udaipur and Computer Society of India, Udaipur Chapter.

The Subject Expert Mr Ashutosh Pancholi (Managing Director, Ashutosh Data Products, Chittorgarh (Rajasthan)) during the technical talk stressed on unleashing the power of

VIRTUALIZATION encompassing technical fundamentals on setup and implementation of Virtualization, with future avenues. The discussion included in detail about Blade Centre Server, Hypervisor, VMware covering the strengths and opportunities in overcoming contemporary business problems.

The seminar provided a platform for in-depth discussion and was highly interactive in nature. The

Patron

Dr. Seema Singh

Chairperson & Managing Director
Aishwarya Education Society

Editor

Dr. Archana Golwalkar

(Director, AIM & IT)

Student Members

Aamir Sanwari

(MCA)

Meenakshi Rao

(MCA)

Ritesh Chouhan

(MCA)

Suman

(MCA)

IMPORTANT LINKS

www.aishwaryacollege.org

www.rtu.ac.in

www.csi-india.org

www.ekalavya.it.iitb.ac.in

www.nbrc.ac.in/library/free_ebooks.htm

www.delnet.nic.in

session was attended by Members from CSI Udaipur Chapter, Students from other colleges, Aishwarya CSI Student chapter Members and staff with great zest for learning.

Eminent Members present were Prof. Dr. Vijaya Laxmi Chouhan (Dean, Aishwarya Institute of Management & IT), Dr. Azimuddin Khan (President CSI Udaipur Chapter), Mr. M L Talesara (Vice Chairman CSI, Udaipur), Mr. Mazhar Hussian (MC, CSI, Udaipur). The program was coordinated by Dr Archana Golwalkar (SBC, Aishwarya CSI Student Branch).



Shri Ashutosh Pancholi delivering on VIRTUALIZATION & CLOUD COMPUTING



Audience learning from subject expert Shri Ashutosh Pancholi

RECENT TECHNOLOGICAL TRENDS AND CAREER OPPORTUNITIES IN ICT

CSI Student Branch, Aishwarya Institute of management & IT, Udaipur (RAJ) Region III

28th April 2012 an interactive guest lecture on "Recent Technological Trends and Career Opportunities in ICT" was organized.

Guest speaker was Dr Azimuddin Khan (Immediate Past President, CSI Udaipur Chapter) Manager Systems at RSMM Ltd. Udaipur, Rajasthan, India. The objective of the session was to make the students aware of recent technological trends and career opportunities available in the field of Information and Communication

Technology in order to prepare them to plan and decide their professional goals by way of specialization. The speaker focused on ASP.NET, Java, PHP, Databases and web development and by means of live examples & case studies, sharing practical approaches of business organizations. This gave the students an insight on how to be a strong entrant by acquiring technical skills with recognized certification in alignment with the corporate requirement. The students' took keen interest in the session and posed queries related to their career path



Dr Azimuddin Khan, Guest Speaker, delivering lecture



10 TIPS TO GO FROM A BEGINNER TO AN INTERMEDIATE DEVELOPER

– Justin James



Having trouble finding tips for beginner developers who want to take their career to the next level? Justin James aims to fill this information gap with his suggestions about how to make that leap.

1: Learn another language

It doesn't matter which language you learn, but learning another language (regardless of how many you already know) will make you a better developer. Even better is to learn one that is significantly different from what you already use on a regular basis. In other words, if you are a C# developer, learning VB.NET or Java will not

help you as much as learning Ruby or Groovy.

And when I say "learn another language," I mean really learn it. Learning a language consists of three realms of knowledge: the syntax, the built-in operators and libraries, and "how to use it." The first two are easy; I think that an experienced developer can pick up enough of a language's syntax

to maintain code in 30 minutes to a few hours depending upon the language. The operators and libraries are just a matter of slowly accumulating knowledge and being willing to check reference materials until you memorize what you need to know. But it's the third item - "how to use it" - that can only be learned over months of working with a language and that's where the real magic happens. I suggest doing a project that is well suited for that language and doing it in that language's style.

Truly learn another language, and I promise that your abilities as a developer will start to blossom.

2: Learn advanced search techniques, tactics, and strategies

More and more, being a good developer is not just about your skill, but your skill at finding information. Simply put, modern languages and development frameworks are too large for most people to remember much of them. As a result, your ability to get work done is often dependent upon your ability to perform research. Unfortunately, knowing how to find accurate, high-quality information is more than just heading to TechRepublic for the answer or typing a few words into your search engine of choice.

"Techniques," "tactics," and "strategies" may sound like synonyms, but they are not. The techniques you need to learn are the advanced search systems of your favorite search engine; you need to learn things such as the Boolean operators, how to filter results (negative keywords, domain restrictions, etc.), what

role word order plays, and more. So essentially, RTFM.

You should learn tactics such as knowing how to approach any particular search and knowing what you should you actually look for. Errors are easy - just look for the error code - but keyword selection on many searches is much more difficult.

With regard to strategies, you need to learn things such as what search engines to use (hint: general purpose search engines are not always the right answer), which sites to visit before going to a general purpose search engine, and even which message boards to post to for help.

3: Help others

Teaching others is invariably one of the best ways to learn anything. It is understandable to think that you don't have much to offer because you are relatively new to the development field. That's nonsense. Remember, everything you know you learned from someone or somewhere; so try being 'the someone' that another person learns from. Spend a few minutes a day trying to answer the questions on TechRepublic or another site as best you can. You can also learn a lot by reading other members' answers.

4: Be patient and keep practicing

Research shows that it takes "about ten years, or ten to twenty thousand hours of deliberate practice" to become an "expert." That's a lot of time. Furthermore, becoming an expert does not always mean doing the same task for 10 years; it often means doing a wide variety of tasks within a particular domain for 10 years. It

will take a lot of time and energy to become an "expert"; working as a developer for a few years is not enough. Want to become a senior developer in your early 30s? Either start your education/training sooner or be willing to do a lot of work,

reading, and practicing in your spare time. I started programming in high school, and I devoted a lot of off-hours to keeping up with the industry, learning new skills, and so on. As a result, I hit the intermediate and senior level developer positions significantly earlier in my career than most of my peers, which translates to an awful lot of money over time.

5: Leave your dogmas at the door

Time for some brutal honesty: Beginner developers probably don't know enough to state that there is One Best Way of doing something. It's fine to respect the opinion of a friend or an authority figure, but until you are more experienced, don't claim their opinions as your own. The simple fact is, if you don't know enough to figure these things out on your own, what makes you think that you know which "expert" is right? I know this sounds really harsh, but please believe me; I have met far too many budding developers who had their careers or their growth set back years because they got hung up on some foolish piece of advice or followed some "expert" who really didn't know what they were talking about. A great example of this is the abuse of object-oriented architecture. For example, many beginners read some information about OO, and suddenly the class diagrams to their simple applications look like the Eiffel Tower.

6: Learn a few advanced ideas in-depth

Much of what goes into being an intermediate developer is having a few concepts that you are really good at working with in code. For me, it is multithreading / parallelism, regular expressions, and how to leverage dynamic languages (and the last two are fading as I get farther away from my Perl history). How did this happen? Multithreading and parallel processing came about because I read articles on it, thought it sounded interesting, and figured it out on my own; I keep writing apps that use those techniques. I had a job that used a ton of regular expressions in Perl. Also, I ended up writing my own e-commerce engine with a template processing engine and built-in database system; then I spent nearly two years working on it.

Find something that has you really hooked. It might be image manipulation or maybe database design or whatever. Even if you're an entry-level developer over all, try to become an expert in at least one area of focus. This will get you into that intermediate level quite quickly, and once there, you will be halfway to expert.

7: Learn the basic theories underlying your field

Its one thing to write "Hello World," but it's another to understand how the words appear on the screen. By learning the "groundwork" that supports the work you do, you will become much better at it. Why? Because you will understand why things work the way they do, what might be wrong when things are broken, and so on. You will become better by learning what happens at a lower level than your work.

If you are a Web developer, read the HTTP RFC and the HTML spec. If you use a code generator, really look at the code it generates; if you use database tools, take a look at the underlying SQL it generates; and so on.

8: Look at senior developers' code

At your job, take a look at the code the senior developers are writing and ask how and why things were done a particular way. If you can, check out open source projects as well. Even if other developers don't have the best coding habits, you'll learn a lot about how code is written. Be careful not to pick up bad habits along the way. The idea here isn't to just blindly imitate what other developers are doing; it's to get an idea of what works and what makes sense and try to imitate it.

9: Learn good habits

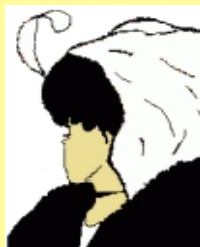
Nothing marks an inexperienced coder like stupid variable names, poor indentation habits, and other signs of being sloppy. All too often, a developer learned how to program without being taught the less interesting details such as code formatting - and it shows. Even though learning these things will not always make your code better or you a better developer, it will ensure that you are not viewed as an entry-level developer by your peers. Even if someone is a senior developer, when variables are named after their 97 cats or their functions are called "do Something ()," they look like they do not know what they are doing, and it makes their code harder to maintain in the process.

10: Have fun

Want to be stuck on the career treadmill? Hate your job. What it takes to move up in this business is not merely dogged determination to bring home an ever growing paycheck but an actual enjoyment of your work. If you do not like your work and you are a junior developer, what makes you think that being an intermediate or senior developer will be any better? Change jobs or change careers. On the other hand, if you love the work you are doing, great! I guarantee that you can become a better developer if you keep at it.

Puzzle

- Q: If you were running a race and you passed the person in 2nd place, what place would you be in now?
- A: You would be in 2nd place. You passed the person in second place, not first.



- Q: What do you see at first glance - an old lady or a young lady?
- A: They are both there! Mouth of the old woman is necklace of young woman and nose of the old lady is chin of the young lady.