



binary

010010010001010101010010101010011110101010

Volume 01 Issue 1.0



Learn . Code . Share

Open Ideas, Open Minds

Binary is the CSE department's technical magazine.

The constant zeal to learn and explore
that defines CSE also defines Binary.

The FireFox Os

Know the story
of FireFox OS

How Web Works

A kickstart for those who want
to start web development

DeepDive - CSS3

CSS tricks that make
Web designing easy

Ruby Vs Python

The rivalry explained

DNA Storage system

The Entire World
in A Shoebox

Deep Science

Learning to learn the future, make a
move to understand **Artificial Intelligence.**

```
function HelloWorld()
{
    myname = "Binary !";
    alert(myname);
    console.log("101010101");
}
```

Data Commodification

A Global Problem?

Quantum Computing

Beginning
of a new era

Fun with Chrome

Cool stuff with Chrome

FreedomBox

A step towards
decentralization

"Good teachers know how to bring out the best in students"

-Charles Kuralt

LETTER FROM THE HEAD OF DEPT.

*Dr. Y Rama Devi,
Head Of Department,
CSE Department,
CBIT*



It is with great pleasure that I am writing this letter for the first edition of the Binary magazine. The initiative taken by students of our department is highly commendable, and will serve the dual purposes of gaining knowledge and spreading awareness.

Binary promises to be a source of information and inspiration to everyone who spends some time perusing its pages. The aim of promoting a culture of open-source technology in college and elsewhere is a noble one, and the work done by students of editorial board, is appreciable in this regard. Also commendable are the efforts of the Binary team to curate and publish articles that they feel are important for every student to know about.

I would like to encourage all our students to follow further editions of Binary, and find between its pages not only knowledge but also the motivation to go out and make a difference, big or small. As engineers, our goal is to constantly innovate and make the world an easier place to live in. Let us all stay smart and strong to achieve it.

LETTER FROM THE FACULTY CO-ORDINATOR



It is indeed a pleasure and an honor to be writing these few words for the inaugural edition of Binary, a technical magazine published by the department of computer science and engineering, CBIT.

It is with a touch of sadness we observe that the reading habit among the present day youth is on the wane. I wholeheartedly hope and believe that this initiative to bring out a technical publication by the students of computer science and engineering of CBIT is going to reverse that trend. We intend Binary to be a part magazine, part technical periodical. In addi-

*V Madhusudhan Rao,
Associate Professor,
CSE Department,
CBIT*

tion to publishing articles relating to the latest happenings in the technology world, it is designed to showcase the achievements of the students of computer science as well. I hope Binary will act a medium to spur our students on to greater creative and innovative results.

Bringing out a quality technical magazine/periodical necessarily demands hard work and passion of the highest order from all the individuals involved. I am sure the founding members have it in them and hope they will pass on the tradition to their successors.

A WORD FROM CBIT OPEN SOURCE CLUB

By Sekhar Karedla

As the founder and convenor of CBIT Open Source Community, I welcome all of you to be a part of this community. I have only one question to ask you: Which is better, a product developed by a handful of developers or a product developed by thousands of developers? There is always a fear among us to start something new, to get out of our comfort zones. Success cannot be achieved while sitting in our comfort zone. We as learning individuals should always expand our comfort zone. Starting an Open Source Community in a college which is dominated by proprietary software was a challenge for us.

The reason my team and I were determined to start an Open Source Community and not just any software club is that Open Source is the future of our technical society. Over the last ten years, the awareness about open source is constantly increasing. Two strong tools we inherit from the culture of Open Source are choice and collaboration.

Open Source technologies are the foundation of many modern applications. A person always learns from their mistakes, but when everyone learns from a person's mistakes, that individual is said to have made a contribution, and this club will provide an environment that ensures that even if you fail at an individual level, you will succeed at a community level.

CBIT has faced many issues regarding licenses and proprietary software. Our hope in starting this club is to at least make a dent in college expenditure on proprietary software and services.

Finally, I thank Dr. Y Rama Devi (HOD, CSE), Dr. K Radhika (HOD, IT), and Dr. M Swamy Das (Associate Professor) for the immense support and encouragement they have provided us.

I would also like to thank the Swecha organization for their technical support and guidance in running the activities of this club smoothly.

MAGAZINE COMMITTEE

Chief Patron

Dr.B.Chennakesava Rao
Principal,CBIT

Chairperson

Dr.Y.Rama Devi
HOD CSE

Faculty Co-ordinator

V.Madhusudhan Rao

Technical In-charge

G.Vivek

Editors-In-Chief

Vihar Kurama
Richa Kulkarni

Layout Chief

Himanshu Kapashi

Deputy Editors

Akshay Desai
Pranith Kumar

LETTER FROM THE EDITORS

Vihar Kurama, Richa Kulkarni

We are thrilled to present to you the inaugural edition of Binary. When the idea for a magazine first came to us, it was a simple aim that we had in mind: to share things that we find interesting with everyone else who might benefit from them. Binary is the perfect platform for such an endeavour – its aim aligning exactly with our motto of “open ideas, open minds”.

The support of the Principal, Head of Department, Faculty Co-ordinator, and Technical In-charge are crucial to our attempts. We thank them for their enthusiasm and guidance. In the coming editions, we shall bring to you some of the finest ideas conceived by people who are changing the world, one line of code at a time. It is our hope that we can all learn from one another, and grow to achieve everything we dream of today.

To contribute your ideas to Binary, please contact us at binarycbit@gmail.com.

'Facebook is not your friend, it is a surveillance engine"

- Richard Stallman

AN INTRODUCTION TO COMPUTERS

By Richa Kulkarni, BE 2/4 CSE-3

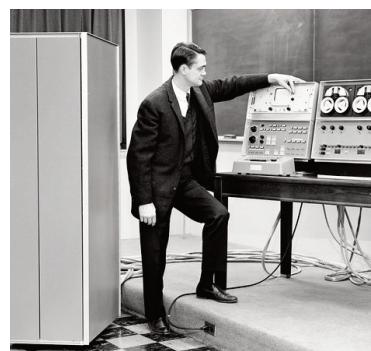
To many of us, the question "what is a computer?" might seem silly at this stage in our education. Surely, without knowing the answer, we could never have come so far. However, a look back at how far we have arrived will only help us decide where to go next.

With this aim, then, let us ask ourselves this question: what is a computer? The vast (and often overwhelming) variety of answers available all means one simple idea: a computer is a machine that receives information from a user, performs certain operations on that information, and gives the results to the user in a useful form. This purpose is achieved through the combination of hardware and software, hardware being the physical components of a computer, and software being the programs that interact with the hardware.

That brings us to our second area of interest: programs. What is programming?

Again, simply put, programming means designing a sequence of steps that will perform a particular task and represent that in the form of instructions that can be understood by the machine. It is important to note here that with the plethora of programming languages available, we would be remiss in focusing on learning only new languages, and not on imbibing new ideas. The study of programming is as much about ideas, logic, and

critical thinking as it is about Java, Python, or Ruby.



Now that we have laid down the original definitions of a computer and programming, we must look at the computer not only as a complex and intricate machine that qualifies to be a work of art but also as the most indispensable tool in our hands today. With increasing digitization and growing technology, it has been rightly said that the future belongs to computers. Let us make sure that we are ready for it.

FreedomBox was built with an idea of eliminating intermediaries. It all starts with the simple concept of data. Millions and millions of data are being transferred every day, and the problem or challenge here is: where our data is being stored? And how is it being used?

One of the best examples is Facebook. Why is it that you receive posts only from one set of friends? The simple answer is that Facebook filters all your information and will only give you the relevant information. That means we only see the information that Facebook wants us to see rather than what we want to see. The danger here is it kills the ability

WHY IS INFORMATION SECURITY IMPORTANT?

By SSV Likhitesh, BE 3/4 CSE-2

News articles these days contain a myriad of information. However, today, something is changing. A lot of news has surfaced regarding stolen bank credentials, stolen CVVs and compromised Facebook accounts for sale. Why is this happening? The answer shall become apparent as we proceed. To understand why information security is important, let us start off with one basic question. "Why security, at all?" The answer is precisely the same as the answer to why one would need a security guard at their door. Now let's look at what this field has in store for us.

Every day, we observe news saying that some or the other company was hacked. Even GMail was hacked recently. Yahoo has been hacked three times straight in 2016. Amazon was knocked out of service in 2016 for more than half a day causing it an estimated loss of a couple of billion dollars. There is a difference between choosing something to be a field of career and having minimum awareness about it. Despite information security jobs currently being the highest paid jobs in the IT sector, let's focus on why a minimum knowledge

is necessary. Any application developed ultimately needs to win the trust of the customer. If your application is proven to have a flaw which allows an attacker to steal your customer's private data, then imagine the consequences. Hence, it is a bare necessity not to just program, but to secure code, which implies that one needs to be aware of security, as one cannot secure or strengthen something unless one knows how to break it. Even in your leisure time, bug bounties are decent earnings that supplement your income.

Finally, even if machines ever become as intelligent as humans, it is common sense to say that a computer can never be smarter than its creator. Which means that the same limitations that apply to people also apply to machines. So, if a person can be fooled, even an equally intelligent computer can be fooled, which practically is the same as gaining unauthorized access. Taking all this into account, it can be said that IT industry has tremendous scope in the field of Information Security, and regardless of how intelligent computers might ever get, this field would continue to thrive.

to think independently. All the Facebook posts are in a central location, and the same is the case with WhatsApp and others.

This means that there is a need to control the flow of our data, and this is where the issue of intermediaries comes into play. It is our data, and there are these middlemen who are using it and manipulating it, performing analytics for their purposes. Most of the services we are using today are centralized. The FreedomBox was built to decentralize these services.

We can consider FreedomBox as a Wi-Fi router with additional services. People who are connected to the network can use

these services. The biggest advantage here is that we own our data. The FreedomBox can also be installed on a Debian-based operating system. Once it is installed the configuration and functioning are similar to that of a smartphone. FreedomBox includes secure instant messaging at low bandwidths, high-quality voice conference calls. It lets you publish your content in a blog and wiki to collaborate with the rest of the world.

At present FreedomBox software has been installed on the campus. It's up and running, and the currently installed services are Mumble (Voice chat) and SIP Server (repro).



FreedomBox

ELIMINATING MIDDLEMEN

By Pranith Kumar, BE 3/4, CSE-2

"The question of whether a computer can think is no more interesting than the question of whether a submarine can swim."

-Edsger W. Dijkstra

HOW THE WEB WORKS

By Vihar Kurama, BE 2/4 CSE-3

Do you want to build a website, but are confused about how to start? Here are the basics that will help you to understand the core concepts of the Web.

Here are the four segments that network works on:

- Part 1: Parts of Web.
- Part 2: Connecting the parts (structure).
- Part 3: Coding the Web.
- Part 4: Painting the Web.

Parts Of Web

Clients are the web user's Internet-connected devices (for example, your phone connected to your Wi-Fi) and Web-accessing software available on those devices such as Chrome and Firefox. Every Client is named after an address known as the IP address that other computers can use to identify it.

Server: A server is a machine that is connected to the internet and also has an IP address. A server waits for requests from other computers (e.g. a client) and responds to them. The main work of the server is to process, store, and deliver web pages to clients.

IP address: An IP address is a binary number that uniquely identifies computers and other devices on a TCP/IP network. IP stands for Internet Protocol. IP addresses have four sets of numbers separated by decimal points (e.g. 244.155.65.2) and this is generally how IP is represented.

DNS: DNS stands for "Domain Name System." The function of DNS is that it keeps track of domain names and their corresponding IP addresses on the Internet.

Domain Name: Domain name generally identifies one or

more IP addresses. For example, if the user uses the domain name "www.google.com" the DNS uses it to look up the corresponding IP address for that given website.

TCP/IP: TCP/IP stands for "Transmission Control Protocol/Internet Protocol." A "protocol" is simply a standard set of rules for tasks. TCP/IP is used as a standard for transmitting data over networks.

HTTP: HTTP stands for "Hyper Text Transfer Protocol." HTTP is the foundation of data communication for the World Wide Web. It is generally a protocol that web



browsers and web servers use to communicate with each other over the Internet.

URL: Uniform Resource Locators, or URLs, identify a particular web resource. A simple example is <https://facebook.com/someone>. The URL specifies the protocol ("https"), host name (facebook.com) and file name (someone's profile page). Here is the connection between all these parts:

First, when you enter the URL (www.google.com) in your browser, the browser parses the information in the URL, which includes the domain name and protocol (for example https), and then the browser pings the index page. Later, the browser communicates with your ISP to do a DNS lookup of the IP address for the web server that hosts

the URL you entered. Once the Internet Service Provider receives the IP address of the destination server, it sends it to your web browser. Your web browser sends an HTTP request to the web server for the main HTML web page of www.google.com.

Coding the Web:

Now let's see how the server automatically updates all the web pages. This is generally known as the backend or "server-side." This refers to everything the user can't see in the browser, like databases and servers. Backend developers should be aware of all the security handling issues and content management. Basically, backend is used for building dynamic websites (where content is automatically rendered time to time). Web developers choose programming languages like Python, .Net, PHP, Node.js etc. to code the backend. Blogs are dynamic sites since their content is constantly changing and updating. A dynamic site requires a database to work properly. The code they write communicates with the server and then tells the browser what to use from the database.

Painting the Web

The process isn't done yet. Here comes the most interesting part of the web – design. Generally, we use HTML (Hyper Text Markup Language) to generate simple content in the website, and to beautify the content we use CSS (Cascading Style Sheets). Later, JavaScripts are added for the interactivity and the liveliness of the pages.

The **WEB** is complicated but you have completed the hardest part!

FUN WITH chrome

By Venkatesh E, BE 2/4, CSE-2

Spend two minutes on this and it will make your day. I'm going to tell you some tips and tricks for Google Chrome which will make you stand out in a crowd.

COOL STUFF #1

Consider this situation. You have your Chrome app opened on your desktop and feel like listening to some music. What any normal person would do is to go to the location of the multimedia file and then play that file using some random player. Now I'm going to tell you what I would do. I simply drag that file into my Chrome window. Tada!

COOL STUFF #2

Ever heard about Momentum theme? The first thing to do after reading this article is to add the Momentum theme for Chrome. What is it like to have a background image, a to-do list, a line in English wishing you by mentioning your name, and a quote when you open a new tab in Chrome. Pretty cool!

COOL STUFF #3

You know that even Chrome has a task manager. You can find it here: More tools → task manager. To make it simpler without much hassle, press SHIFT+ESC and a beautiful task manager opens up.

COOL STUFF #4

Can't stare at the white background of any website during nights? Yes, I'm going to tell you about an extension: Dark Reader. You can switch to dark mode easily by adding the Dark Reader Chrome extension. It is customizable. You can change brightness, sepia, grayscale, contrast levels. Simple and cool!

That's all folks!

"Information is power. But like all power, there are those who want to keep it to themselves"

- Aaron swartz

CSS TRICKS THAT MAKE WEB DESIGNING EASY

By Sriram Veeraghanta, Head of Design, Tutor.co

1. Variables in CSS:

In web designing every front-end developer uses some different color codes in websites. If you are using CSS instead of SASS while building a website you need to specify the color code whenever you are using it.. If you came across a situation where, for example, you need to change the website theme, it is very hard to change each and every line in which you have used that color. In such circumstances, CSS root will make your work easy. Here you are going to specify the color codes once inside your styles.css file and reuse them inside the file wherever you need them.

The following code explains how to use CSS root.

```
...
:root {
  --primary-color: #F44336;
  --secondary-color: #B71C1C;
  --text-color: #333333;
}
h1 {
  color: var(--text-color);
}
div {
  border: 1px solid var(--primary-color);
}
button {
  background-color: var(--primary-color);
}
button:hover {
  background-color: var(--secondary-color);
}
...
```

For simplicity, we can say that we are declaring some CSS root variables for reusability.

2. Color Filter over an image:

While we are working with images as backgrounds, which is predominantly used in websites, some people call it a hero-unit. Here we will have an image as background and text over it. Some

images need to have a black color layered film on the image to make the text visible. Web designers use Photoshop to make a film over the image, but it takes a lot of iterations to make it a perfect hero-unit. In such cases, we use a CSS linear gradient over the image using the following snippet.

```
...
.hero-unit-div {
  background: linear-gradient(rgba(0,0,0,0.7), rgba(0,0,0,0.7)), url(https://static.pexels.com/photos/343299/pexels-photo-343299.jpeg) no-repeat center center;
  background-size: cover;
}
...
```

3. Object Fit Image:

Image tags are good enough in HTML when you are using the background for divs by using `background-size: cover;`. We make the background image scaled and fitted inside the div correctly, but the image tag does not have any such image scaling properties. In CSS3 a new feature was introduced for making the images inside images to scale as a cover. The following snippet explains the Image tag cover.

```
...
img {
  object-fit: cover;
}
...
```

These are the some of the CSS tricks I found while experimenting with different designs. So give it a try, if you haven't already. Maybe it will be fun; maybe you will even learn something new by experimenting with these tricks. The important thing is to make use of all the hidden features of CSS, which makes your work straightforward and fun.

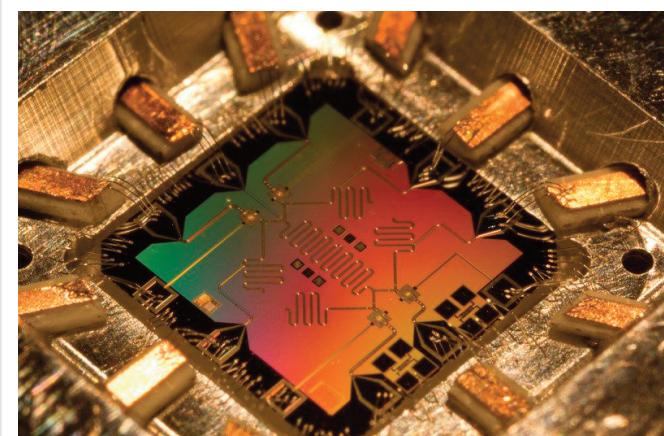
Merry CSSing!

QUANTUM COMPUTING BEGINNING OF A NEW ERA

By Revannth Vedala, BE 2/4 CSE-1

Computers use classical bits in processing their operations. These bits have the capacity of holding either a 1 or a 0. However, lately, a third possibility is being debated and deliberated upon: what if a bit could hold a 1 and a 0 simultaneously? These hypothetical bits are called qubits. Don't judge them yet, these bits may still be in theory, but several researchers have validated their possible existence in the immediate future, so much as to predict their creation to occur in the next 10-12 months. Though this might be a little far-fetched, qubits promise the beginning of a new era, and with them they bring an era of highly advanced computers, the Quantum Computers.

The hype around this technology might be a little difficult to understand for a day-to-day computer user, but we promise that by the end of this article, you will share the same excitement about them as the world does. First and foremost, the advantage of a quantum computer is the ability to scale its performance exponentially with a negligible increase in qubits. It is said that a 300 qubit computer could do more calculations than there are atoms in the universe. Not just that, with quantum computers, Artificial Intelligence will see a rapid growth which in turn will pave the way for the so-called "gadgets of the future."



Apart from the advantages of scaling in performance, we get an added benefit of downsizing. An easy way to explain it would be by giving a real time example: the world's most powerful computer (whose size is equivalent to that of half of a football field) could fit on a small chip of 50-60 qubits. That would enable our smartphones to have the functions of a super computer.

The race towards quantum computers has already begun. Many big players like Google have claimed to create these computers by the fall of next year. This race will definitely impact the dynamics of the world. It will surely give us a means of exploring and understanding our complex universe, but at the same time will open doors to unforeseen threats.

"People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world." — Pedro Domingos

THE ENTIRE WORLD IN A SHOEBOX – DNA STORAGE SYSTEM

By G. S. Harsha Anirudh, BE 3/4 CSE-2

Everyday millions of pictures, videos, and other multimedia are stored on hard drives and secondary storages. It is estimated that around 600 TeraBytes of data are stored only on Facebook servers every day. Imagine how much data is stored on the servers of all companies, and how much data is stored in a year! The amount of data stored exceeds our imagination. The place required to store this humongous data requires warehouses of millions of square feet and with huge maintenance costs.

Scientists have come up with a solution: storing the data in human DNA. This idea of recording, storing and retrieving data from DNA was first given

by Mikhail Neiman in 1964 in the Radiotekhnika journal. On August 16 2012, George Church and colleagues at Harvard University published a journal which described how DNA was encoded with digital information, like an HTML draft of a 53,400-word book,

11 JPEG files, and a JavaScript program. They showed that 5.5 Petabits can be stored in a cubic millimetre of DNA. They used a very simple code for retrieval which is usually quite error-prone. This was the very first instance which proved that DNA

can be used in place of traditional storage devices.

DNA data storage uses a base sequence of nucleic acids. It uses oligonucleotide synthesis



machines for storage and DNA sequencing machines for retrieval. DNA is much more compact and is perennial compared to hard drives,

and also has resistance to obsolescence. Tapes use 0's and 1's to store data, where the nucleotides adenine (A), cytosine (C), guanine (G) and thymine (T) constitute a strand whose combination is used for storing the data. It is much better than

tapes in terms of longevity because of the three-dimensional packing in DNA, which isn't possible in transistors because of heat issues.

The major drawback here is that the read latency is much higher than the write latency, as the entire pool must be sequenced to read a small block of data. Selective DNA amplification with polymerase chain reaction (PCR) technique can be used to access a block of data, but the cost of using this cutting edge technology is too high. Recent research shows that data can last centuries when kept in suitable environment, and all of the data present in the world today can be packed in a single kilogram of DNA.

related to Internet of Things.

Each group was assigned a mentor and the discussion went on for about three hours. It was a great opportunity to interact with others and the working lunch we had gave a taste of how things work in the office. Then each group presented their product and the creativity and ideas that went into their presentation show why we need more women in technology. The judges helped us introspect regarding any improvements that could be made.

In the end, each participant was awarded a certificate and received amazing goodies. The winners in the quiz and the activity received great prizes, and then it was time to leave. The really hospitable people and the impressive infrastructure of their campus made it even harder to leave. It left us looking forward to attending more such events in the future.



By Krishna Sri, BE
2/4 CSE-3

job any better. About 45 women from all over India attended the event and the team took care of the travel and accommodation. They had a fun yet productive and engaging day planned for us. We were also joined by the women working there who mentored us diligently.

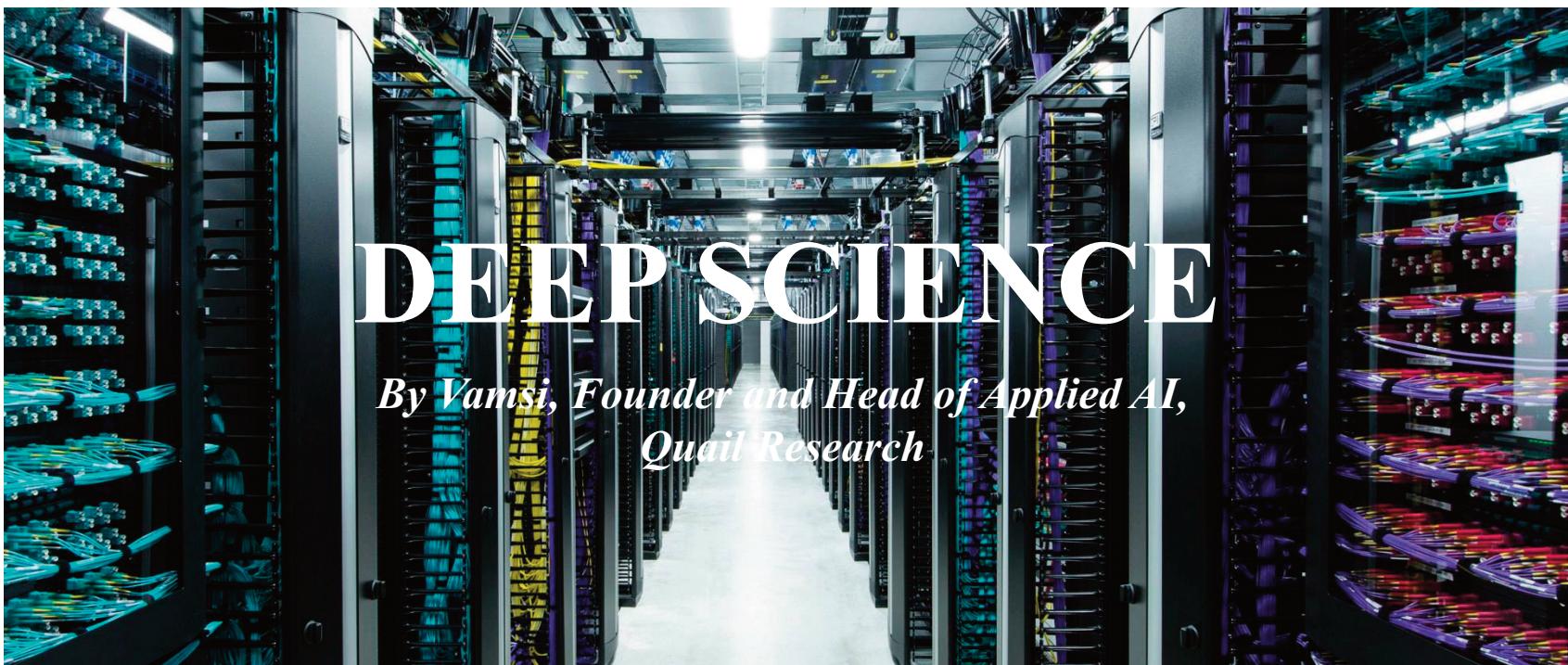
The event started off with a timed quiz just to break the ice and then we had Charumathy Srinivasan, Partner Group Engineer Manager in Microsoft India Development Center, deliver a very inspiring speech on

her journey since she started working at Microsoft in 1992. She spoke about the principles that helped her reach where she is now and the technologies she works with. She also talked about the growth mindset they embrace and their mission to empower every person and organization on the planet to achieve more. We were then briefed about the group activity that will be going on for the rest of the day – each group was supposed to design and present a product based on two keywords

Codess is an initiative by Microsoft to inspire female coders and to help them achieve their professional goals through various events and workshops worldwide. This particular event was organized by the University Recruitment Team of Hyderabad and no one could have done the

'A computer would deserve to be called intelligent if it could deceive a human into believing that it was human "

- Alan Turing



Artificial Intelligence, Machine Learning, Representation Learning, Deep Learning – these are the most buzzing words in the industry right now. So, what are these? It all started with a simple question: "Can machines think?" Inventors have long dreamed of creating machines that can think. This desire of creation dates back to at least the time of ancient Greece.

Machine learning deals with the — problem of extracting features from data and connect the dots between them so as to solve many different predictive tasks:

Forecasting (e.g. Energy demand prediction, predicting stock prices, finance)

Inputting missing data (e.g. YouTube recommendation, Netflix recommendations)

Detecting anomalies (e.g. Security, fraud, data leakage, virus mutations,)

Classifying (e.g. Credit risk assessment, cancer diagnosis)

Ranking (e.g. Google search, Facebook feed, Instagram feed personalisation)

Summarising (e.g. News summaries, social media sentiment analysis)

Decision making (e.g. AI, robotics, compiler tuning, trading, sometimes even driving a car (Tesla Autopilot))

When programmable computers were first discovered, people wondered whether such machines might become intelligent – over a hundred years before one was built. Today, artificial intelligence (AI) is a growing field with many practical applications and current research problems. We look to intelligent software for the automation of routine labour, recognising speech or images, making diagnoses in medicine and support-

ing fundamental scientific research.

In the early days of artificial intelligence, the field rapidly undertook and solved problems that are intellectually difficult for human beings but relatively a very straightforward model for a computer. Problems that can be described by a list of formal, mathematical rules like your general computer programs, but these worked upon repeated data and can be applied to many problems.

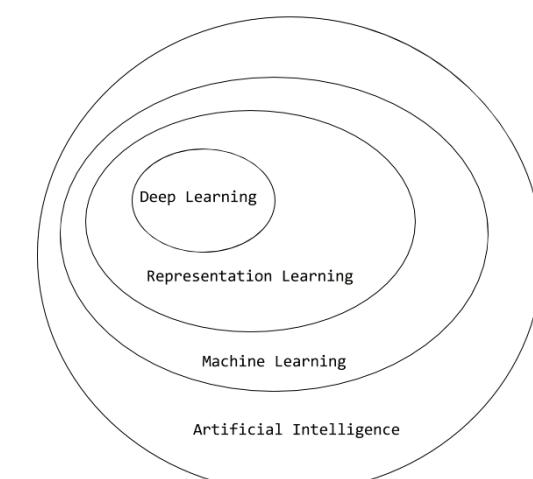
The actual challenge in the field of artificial intelligence proved to be solving the native tasks that are easy for people to perform naturally but very hard for people to describe formally. Problems that we solve intuitively, that feel automated, like recognising spoken words, predicting stock prices, faces in images, or something predicting most of the things on the planet.

The difficulties faced by systems relying on hard-coded knowledge suggest that AI systems need to attain the ability to acquire their own knowledge and behold intelligence with them, by extracting patterns from raw data. This capability is known as Machine learning. A simple machine learning algorithm called Naive Bayes can separate legitimate e-mail from spam e-mail.

The introduction of machine learning allowed computers to tackle the problems that involved knowledge and intelligence of the real world and make decisions that appear subjective. For example, when the regression problem is used to recommend delivery, the Artificial Intelligent (AI) system does not analyse the patient directly. Instead, the doctor tells the system several pieces of related and relevant information, such as the

presence or absence of a scars or abnormality in the organ. Each piece of information included in the representation of the patient's symptom is known as a feature. The regression (one technique of ML) learns how each of these features of the patient correlates with various outcomes. However, it cannot influence the way that the features are defined in any way. If logistic regression were given an MRI/CT scan of the patient imaging, rather than the doctor's report, it would not be able to make useful predictions. Individual pixels intensities (an image internally by a computer is an n-dimensional matrix, each element having the corresponding pixel value) in an MRI scan have a negligible correlation between, with any complications that might occur during delivery.

If you want to understand correlation between AI/ML/RL/DL, no knocks here is a Venn diagram for you.



"Karma of humans is AI"

— Raghu Venkatesh

BLOCK CHAIN : AN OPEN ECONOMY

Say we were sitting in a cafe and I gave a document to you. I physically gave it to you and you took it. You know that you may give something to your friend as you wish. Now imagine a scenario in which I gave a digital document to you over some service provider. What is the proof that I did not give that document to my friends, or that I didn't save copies of it with me? What are the chances that the so-called service provider did not take copies of it or that the exact document that was sent was received by you? This mistrust is the problem with a middleman being the provider of a service. Nobody knows what is happening with the document except for the owner, nobody will have any clue if the owner decides to add new documents. This is the working nature of proprietary services.

Imagine a system where a ledger is maintained to count the number of documents being made and sent. Now we can keep track of the documents. What if the ledger is made open to everyone? Every

other user in the network will have the same ledger and any differences in ones ledger can be sorted out (means no cheating!). If you add new documents by solving a problem, you will be awarded with some documents for yourself for the effort you've put in.

Similar to the above story, the centralised service is the present banking system where no one actually knows what is happening. The ledger solution in the story was given by BlockChain.

What is BlockChain?

"Its a software solution that keeps track of all the digital assets of users", by definition.

How does BlockChain work?

1. Transparency: All the users will own the details of every other transaction happening in the network, which basically adds trust to the system.

2. Openness: It is not controlled by a single authority or person. The power of

BlockChain lies in the potential number of users there are in a network. It follows the rules of opensource.

3. Security: Each transaction is done cryptographically with strong encryption. The transaction will pass through every adjacent block (shown in figure).

4. Cryptocurrency: BlockChain uses the digital currency known as Bitcoin, which can be encrypted using cryptography. The document in our story is bitcoin (digital currency).

5. Earning coins: In order to earn new coins you would need to solve problems which are mathematically very tough and algorithmically hard. New coins will be added only when problems are solved.

The advantage of using BlockChain and bitcoins is the efficiency that this technology can offer with the added advantage of transparency. The world is converting to a digital space, and its time we shifted our economy to an open, reliable, and secure digital platform.

By P. Jitendra Kalyan, BE 3/4 CSE-2

DATA COMMODIFICATION A GLOBAL PROBLEM

By P. Jitendra Kalyan, BE 3/4 CSE-2

History has proven that the race for supremacy has not ended at any point, and will not any time soon. As always, the struggle to be at the top was started by the desperate need for a commodity. Be it the colonial attitude that led to medieval imperialism, or the nuclear race that triggered major events in world politics, the reason was always a commodity.

A similar scenario that is taking over the digital era is the hunt for the commodity of "Data". Now the question arises, has data became a commodity? A straightforward answer would be "yes". When you look at the statistics of the past ten years, the graph of online users shot up exponentially. Imagine the amounts of data that are being generated and stored! After processing this data, exceptional results were achieved, and now the variety of services provided over the web is proof that data is the key commodity that is shaping the digital world.

However, is data being commodified a global problem? Data being commodified ranges from our personal and professional data to behavioural patterns on so-

cial platforms and E-commerce sites. Information produced by analysing this data had opened tremendous opportunities for monopolists to take over the digital market.

You may wonder how this would affect an individual. To begin with, the ones in the control can take over the digital freedom of users either directly or indirectly. Directly, by restricting free access to the web. Indirectly, by not letting user decide what they want, or by service providers putting out only information that they want users to access.

This can be understood as one form of slavery – digital slavery to be honest. There was a controversial statements made by an American economist once: "For a nation which was under imperialistic rule for nearly 200 years, losing digital freedom will not be such an issue".

Its high time we look into the problems of data commodification and spearhead the movement against the curbing of digital freedom by multi-national giants such as Google, Facebook, and Uber.

LEARN BY EXAMPLES

By Revannth Vedala, BE 2/4 CSE-1

Programming has become one of the most essential skills to learn these days and so has the need to teach it. Several attempts have been made to teach coding but nothing is as effective as CodeByExamples.

As the name suggests, CodeByExamples uses a unique approach of "learning by experiencing", which promotes a sophisticated mode of learning. This enables an amateur to not only pick up a new language but also ensures that he/she retains it for a longer period of time.

In addition to the simple and eloquent explanation to every example, the website has an indulging user experience which makes the programmer want to stay on the website all day. The user interface is super immersive and the learning process is properly segregated

into a gradual concept building structure. This ensures that the learner stays in the loop of learning and never gets lost. CodeByExamples promises to keep you addicted to the languages provided and will prove to be a valuable addition to your arsenal

of knowledge. The site currently provides four basic languages which integrate the concepts you learn along with the concepts available in the world today.



Today, our thirst for advanced systems has increased immensely and so has the need to design it. CodeByExamples not only ensures a boost to one's resume but also increases one's mental aptitude which is one of the most important skill sets required today. So what are you waiting for, start coding today!

"I do not fear computers. I fear lack of them."

"— Isaac Asimov"

THE STORY OF FIREFOX OS

Not many are familiar with Mozilla's own operating system, Firefox OS. The one-of-a-kind, entirely open-source operating system made the news one last time recently when Mozilla announced its discontinuation. Mozilla, one of the world's largest free software communities, has decided to redirect all its energies towards their Firefox web browser. The story of the Firefox OS, however, is one worth telling.

In July 2011, Mozilla announced its "Boot to Gecko" project (B2G), with the aim of creating a "a complete, stand-alone operating system for the open web." This, in 2013, was renamed the Firefox OS, after their popular browser. The most striking feature of this project was that it sought to create a full-

fledged operating system using only web technologies. Indeed, in its final form, the Firefox OS used only HTML5, CSS, JavaScript, and C++. The four co-founders of the project were Andreas Gal, Chris Jones, Brendan Eich (also the creator of Javascript) and Mike Shaver.

They built an empty GitHub repository and put together a small team to work with the Linux kernel and the existing Android Hardware Abstraction layer. The uniqueness of this project lies in the fact that it eliminated the entire layer of Java code that the Android OS could not function without. In February 2012, the team pre-



sented a prototype at the Mobile World Congress in Barcelona. Here, it gained new partners, and promised to have commercial devices ready to ship the following year. It accomplished this,

becoming arguably the only OS to be visualized and marketed in only two years.

While the developers kept the user interface as simple and as close to Android as possible, they were anxious not to be seen as a third alternative mobile app platform to Google and Apple, but as a new idea that could harness the endless capacity of the open web. After a response to the first version that exceeded expectations greatly,

Mozilla began work on the \$25 smartphone for people in developing countries. After this product failed to take off, the B2G project took a downward turn and began to fizzle out. Many attributed this not to their inability to catch up with Android, but to the lack of effective product leadership inside the company.

The downward spiral continued until Mozilla finally pulled the plug and discontinued its mobile OS, branching into tablets and smart TVs. However, Firefox OS was, and will always be, a fine example of the power of open-source technologies, and as a strong competitor to the two titans: Google and Apple. It was a commendable attempt in the history of app and web development.

By Richa Kulkarni, BE 2/4 CSE-3



RUBY VS. PYTHON

By Prashanthi Paladugu, ReportGarden



Comparing these elegant dynamically typed languages is quite magical. Yukihiro Matsumoto was the first Rubian who created Ruby in the year 1995, whereas a Dutch programmer Guido van Rossum authored Python in the year 1991. Python and Ruby are the best examples of next generation programming languages with tons of out-of-the-box features for web development and with a diverse community with big ties to Linux as well as the academia.

Ruby, when it comes to context of the web, has tons of uses. Its web framework, Rails, has made advanced strides and has huge capabilities for challenging problems, while the community is also huge. Companies like AirBnB, GitHub, and Twitter use Ruby on Rails for development. When it comes to Python for web development, its attention is towards Flask and Django. Companies like Uber and Instagram use this framework in their

production environments.

Cross-sectionally, both the languages have their own advantages. It is likely that people's choices tend more towards Ruby than Python due to the great community of support and the passionately maintained libraries. Python primarily nooks into the data-scientific computing stream a.k.a. the field of Data science.

Languages are born with the intention to serve a purpose, and are not easily made. It's not that Ruby is more powerful than Python or vice versa. It's the use-case that plays well with the language. Before starting a project, as confessions made by creators or communities of languages tell us, never compare apple and oranges. It's the power of the language and its creation that drives the whole web and gives us the power to compute.

Happy Programming.

PYTHON – SCIENTIFIC COMPUTING PACKAGES

By Vihar Kurama, BE 2/4 CSE-3

Analyzing and manipulating data is a big challenge nowadays, and Python's SciPy (Scientific Python), an open source software for scientific computing in Python is now making things easier for playing with data. Here are some of the core packages of Python for data analysis.

1. NumPy: NumPy is a blazing fast maths library for Python with a heavy emphasis on arrays. It is the core python package for scientific computing. It provides various multi-dimensional array objects and an assortment of routines on arrays; it also includes functions for sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, random simulation and much more.

2. Matplotlib: This is a plotting library which produces different quality of graphs in a variety of multiple formats and interactive environments across several platforms. Some of the most popular uses for Matplotlib, especially in recent years, are data processing, analysis, and visualization.

3. Pandas: This high-performance Python subset has easy-to-use data structures and data analysis tools for the Python programming language. One of the most powerful features of

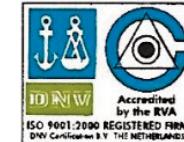
Pandas is that it includes data frames, a data structure similar to spreadsheets or SQL tables. There are several functions in Pandas that can be used on data frames, which make things simpler and faster.

4. SciPy Library: SciPy is packed with mathematical algorithms and utility functions built on the NumPy extension of Python. With SciPy an interactive Python concourse becomes a data-processing and a system-prototyping environment, rivaling systems such as MATLAB, IDL, Octave, R-Lab, and SciLab.

Here are the commands for installing these packages. This requires pip (Python install packages) If your system doesn't have Pip, use the following commands for installing pip in Windows:

1. First download get-pip.py file and run it from the shell using the command "python get-pip.py". In Ubuntu use the following commands "apt-get update", "apt-get -y install python-pip".

2. Now install the following packages:
pip install numpy
pip install pandas
pip install matplotlib
pip install scipy



binary

Open Ideas, Open Minds

The Department of Computer Science Engineering has brought out the very first issue of Binary, CBIT's technical magazine for computer science students. The magazine attempts to share and encourage an interest for ideas that really matter. The ideas we speak about are changing the world, one line of code at a time. Our endeavour to spread the word about open source technologies is reflected clearly in Binary.

COSC

A NEW BEGINNING

*By M Dasarada Ram Reddy, BE
3/4 CSE-2*



Learn . Code . Share

COMMUNITY AGENDA

After the formation of the COSC club, a jump-start was needed to introduce Open Source technologies. Our thought was to introduce Python, a powerful programming language which has a lot of applications. It is used in web programming, networking, and desktop applications among other things. Nowadays it is used as a substitute for conventional programming languages like C/C++ and Java.

Python, an interpreted language has a design philosophy which focuses on code readability and a syntax which allows programmers and beginners to express their algorithms or concepts

in fewer lines of code than would be necessary in languages such as C++ or Java.

In CBIT, students are trained in C/C++ in their first year and Java in their second. It is difficult to develop larger applications such as web based applications and desktop applications in C/C++. In view of this, we decided to introduce Python programming on behalf of our club as our first step. We organised a two day workshop named "Introduction to Python" on the basics of Python on 5th and 6th January 2017, to which there was a response of about 200 students from CSE and IT departments. It was a

hands-on workshop which also had lab sessions for students to practice in.

The sessions were taken by Ganesh Katrapati, Research Assistant at IIIT Hyderabad on both days. At each and every step of our progress, Swecha organisation helped us and played a key role. The workshop was a great success, which was our first and foremost step as a club, thus laying its foundation very strongly. As a club we are proud to announce that the computer networks lab which has been implemented in C till now, has given an opportunity to students to implement the tasks in Python.

- **GIT** : Initiating college level github organisations to spread awareness about its benefits and utilities.
 - Organising two hands-on workshops per semester aimed at training students to build open source tools and applications
 - **Guest lectures** : Inviting experts in various fields to conduct guest lectures for students.
 - **Project Cells** : Forming project cells with the purpose of enabling teams of students to solve real world problems through applications .
 - **Hackathon** : Conducting intensive one day hackathons for building open source applications.



Team COSC

Back row: (L-R) P Jitendra Kalyan, Jeevan Gandla, R Pranith Kumar, Sekhar Karedla, Himanshu Kapashi

Front row: (L-R) Nymisha Kandi, Alekya Reddy, Richa Kulkarni, Vihar Kurama, Dasarada Ram Reddy

