

```
[julia> length("hello world")  
11
```

```
[julia> 21 + 4.5  
25.5
```

```
[julia> 1 + 2  
3
```

Faculty of Computer Applications & Information Technology

```
[julia> if a > 10  
    "bigger"  
else  
    "smaller"
```

julia

A Programming Language

```
 _ _ _ _ _  
 _ _ _ _ _ | Documentation: https://docs.julialang.org  
(_) | ( ) ( ) |  
 _ _ _ _ _ | Type "?" for help, "]"? for Pkg help.  
|_|_|_|_|_|/ _`| |  
|_|_|_|_|_|( _| | | Version 1.0.0 (2018-08-08)
```

Issue 18, June 2021

```
[julia> println("Hello, World!")  
Hello, World!
```

```
julia> █
```

Message From Dean, FCAIT



A group of mathematicians and computer scientists started to work on Julia programming language in 2009, with an objective to offer the advantages of both the worlds, namely ease of programming of high level languages like FORTRAN, Python for complex level numerical and scientific computing and fastness of programming language like C with dynamism of Ruby. The first initial launch took place in February, 2012 so it can be called just 9 year old young language, though many versions have been released, latest being v1.6.2 in July, 2021. The most important, it is MIT certified free and open source, with a liberal license, whose source code can be availed on GitHub.

Julia offers a unique combination of natural syntax and speed, thus solving problem of programming in two languages. (Writing a program in high level language for convenience and rewriting time consuming parts of the code in fast language like C to expediate the process). Though originally created for numerical/scientific computing, it is now a general programming language having many strong characteristics, some of them being implementation of generic functions, dynamic multiple dispatch, parallel and distributed computing, ability to call other programming language packages for example from Python, R whenever need arises. Through multiple dispatch, programs for scientific computing can be written in a natural way. It also has very efficient libraries for Linear Algebra. Julia has a macro system to generate custom code and other metaprogramming facilities which enhances its performance in execution.

In a nutshell, it can be called the language having everything, a developer would ask for. Needless to add, it is increasingly becoming choice of researchers in the field of Data Science, Visualization, Machine Learning, Artificial Intelligence and so on.

-- Dr. Savita Gandhi

From Editorial Desk

"We are greedy, We want more" – Creators of Julia.

With this thought, we are very glad to bring you the next vivid issue of D-Kosmos. For several years we have been using Python, Matlab and other programming languages for creating dynamic websites but here we have gone through the new programming language "Julia", that's open-source, with a liberal license. It can be used for creating scientific computing, machine learning, data mining, large-scale linear algebra, distributed and parallel computing language.

This issue also gives a glance of FCAIT events such as various Academic Activities like Code Express #10, Tech Talk, Webinars, Alumni talk, Cyber Sadez, etc. Even in this pandemic situation, the enthusiasm of students has only increased as they have taken part in various Hackathon events and won exciting prizes in some as well.

Cultural Activities such as Dance, Music, Painting, Rangoli, etc. are done in virtual mode for the students and they have showcased their talent and enjoyed the event.

Though we have taken every effort to keep the magazine as per the expectations of our readers, at the same time we request them to give suggestions or feedback for us at dkosmos@glsica.org.

Chief Editor

Dr. Tripti Dodiya

Members

Dr. Disha Shah, Dr. Poonam Dang
Prof. Bharti Shah, Prof. Monica Gupta
Prof. Garima Mishra

Designer

Prof. Bharti Shah

JULIA: WE WANT MORE

“We are greedy. We want more”, were the exact words used by the creators of the Julia Programming. It was created in 2009 by MIT researchers Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman. Julia was envisioned as a fast and high level open-source computing language.

In 2012, the computer language made its debut to public and has since climbed up the ranks as one of the world’s most popular programming languages, to the point that many programmers now see it as a potential rival to Python, a stalwart language.[1]

Julia Programming Language!

Julia is a high-performance, high-level, and dynamic programming language that leans towards technical, numerical, and statistical computing. It has also become a popular language for everyday or general programming, as well as rapid web development. Many people describe Julia as having a combination of C’s powerful performance and the rapid prototyping capabilities of Python.

The idea of Julia came about when its creators realized that while scientific computing requires high performance, domain experts usually use slower, more dynamic language for daily programming. While Julia’s creators understood that there are several reasons why programmers prefer dynamic languages for certain applications, they still sought to leverage modern language design and compiler techniques for a single environment that would be powerful enough for not only prototyping but also has the efficiency for performance-intensive applications.

In the words of founder of Julia Computing: “We want an open-source language, with a liberal license. We want the speed of C with the dynamism of Ruby. We want a language that’s homoiconic, with true macros like Lisp, but with an obvious, familiar mathematical notation like Matlab.

We want something as usable for general programming as Python, as easy for statistics as R, as natural for string processing as Perl, as powerful for linear algebra as Matlab, as good at gluing programs together as the Shell. We want it interactive, and we want it compiled.”

Growth of Julia

It seems like an odd choice of name for an advanced programming language, but it’s tremendously popular. Julia is now backed by more than 700 active contributors. It also has 1,900 registered packages, 41,000 GitHub stars, 2 million downloads, and a download growth annual rate of 101 percent. Furthermore, more than 700 universities and research institutions use Julia, along with organizations such as Capital One and Netflix. [1]

Among Julia’s many claims to fame include its membership in the “petaflop club,” which means that the computing language used over 1.0 million threads, 650,000 cores, and 9,300 Knights Landing (KNL) nodes to reach 1.5 petaflops per second as it cataloged 188 million stars, planets, and other astronomical bodies. The feat was achieved using the world’s fastest supercomputer in a staggering 14.6 minutes—proof of the language’s performance. [1]

Today, Julia is also the computing language used to power 3D printers, self-driving cars, specialized applications in augmented reality (AR), machine learning (ML), risk management, genomics, and precision medicine. While Julia still has some ways to go before it edges out Python in terms of popularity, it is still in a top 50 language in the Tiobe index.[1]

Why Julia?

Julia works well for scientific computing but is not restricted to just that, as it can also be used for web and general purpose programming. Some of Julia's features are mentioned as follows:

1. Julia is a modern, expressive, high-performance programming language designed for scientific computation and data manipulation.

2. It is designed for distributed and parallel computation.
3. Julia provides an extensive library of mathematical functions with great numerical accuracy.
4. Julia gives the functionality of multiple dispatch. Multiple dispatch refers to using many combinations of argument types to define function behaviors. Julia provides efficient, specialized, and automatic generation of code for different argument types.
5. Julia's ecosystem is not as mature as C++, Python or R's but the growth rate of the penetration of the language is increasing.
6. The Pycall package enables Julia to call Python functions in its code and MATLAB packages using the MATLAB.jl package. Functions and libraries written in C can also be called directly without any need for APIs or wrappers.
7. Julia provides powerful shell-like capabilities for managing other processes in the system.
8. Unlike other languages, user-defined types in Julia are compact and quite fast as built-ins.
9. Scientific computations make great use of vectorized code to gain performance benefits. Julia eliminates the need to vectorize code to gain performance. De-vectorized code written in Julia can be as fast as the vectorized code.
10. It uses lightweight green threading, also known as tasks or coroutines, cooperative multitasking, or one-shot continuations.
11. Julia has a powerful type system. The conversions provided are elegant and extensible.
12. It has efficient support for Unicode.
13. It has facilities for metaprogramming and Lisp-like macros.
14. It has a built-in package manager (Pkg).It's free and open source with an MIT license.
15. It has elegant and extensible conversions and promotions for numeric and other types.
16. It provides powerful shell-like capabilities for managing other processes.

Niyati Mehta, SYBCA

General comparison with Julia

As Liam Tung discusses in an article published in 2020, that Julia is a zippy programming language that has roots at MIT, has published the results of its 2020 annual user survey. The study aims to uncover the preferences of those who are building programs in the language. This year, the survey attracted 2,565 Julia users and developers, up from 1,844 participants in 2019. Python, a language that's developed a strong affinity with data scientists for machine-learning applications, is overwhelmingly the language that Julia developers would turn to if anyone needed another language.[2]

Regardless of popularity index, Python is in the top three programming language and its popularity is being driven by data scientists and a growing demand for machine-learning applications, plus a wealth of Python modules that helps extend its use in various fields. But Julia, which developer analyst firm RedMonk has rated as a language to watch, does have decent support behind it too. Besides Julia Computing, the commercial side of the language, there is the Julia Lab at MIT's Computer Science and AI Laboratory (CSAIL) and an open-source community criticize for its long-term success.

Last year, 73% of Julia users said they would use Python if they weren't using Julia, but this year 76% nominated Python as the other language. [2] "The more experience people gain with Julia, the less they want to use anything else. That's why Julia has been rising rapidly in the IEEE Spectrum rankings and is now among the top 20 languages." While Python rose as the top alternative language for Julia's user base, Viral Shah, co-creator of Julia, co-founder and CEO of Julia Computing argues that Julia users are actually less inclined to switch to other languages as they learn more about Julia.

MATLAB, another Julia rival in statistical analysis, saw its share of Julia users as a top alternative language drop from 35% to 31% over the past year, but C++ saw its share on this metric rise from 28% to 31%. Meanwhile, R, a popular statistical programming language with a dedicated crowd, also declined from 27% to 25%. [2]

Some of these trends look positive for the long-term survival of Julia despite the threat posed by Python as the go-to language for data scientists.

The most frequently used languages after Julia are Python, and then Bash/Shell/PowerShell. And if Julia, which emerged in 2012, didn't exist, most Julia users would be using C++, MATLAB, R, C, Fortran, Bash/Shell/PowerShell and Mathematica. Julia users also revealed what they love and hate about the programming language, which Julia's supporters claim is faster than Python and R for big-data analysis using CSV files for tasks like looking at stock-price states and analyzing mortgage risk.

Table 1 shows the comparison chart with some of the most programming languages. [3]

Table 1. Comparison Chart

Language	Intended Use	Object-Oriented	Functional	Generic	Companies
Julia	General, technical computing	Yes	Yes	Yes	N26, Flitto, Amber by inFeedo, Chai
Python	Application, general, web, scripting, artificial intelligence, scientific computing	Yes	Yes	Yes	Instagram, Spotify, Netflix, Uber.
R Programming	Application, statistics	Yes	Yes	No	NIIT Tech, Oracle Financial Services Software, Paytm, Snapdeal.
MATLAB	Highly domain-specific, numerical computing	Yes	No	No	Broadcom, Diffbot, ADEXT, Volvo Cars
Rust	Application, server-side, system, web	Yes	Yes	Yes	Dropbox, Sentry, Klarna, Brilliant., Postmates.
GO	Application, web, server-side	No	No	No	Soundcloud, Goggle, Uber, BBC, Medium, Dailymotion.

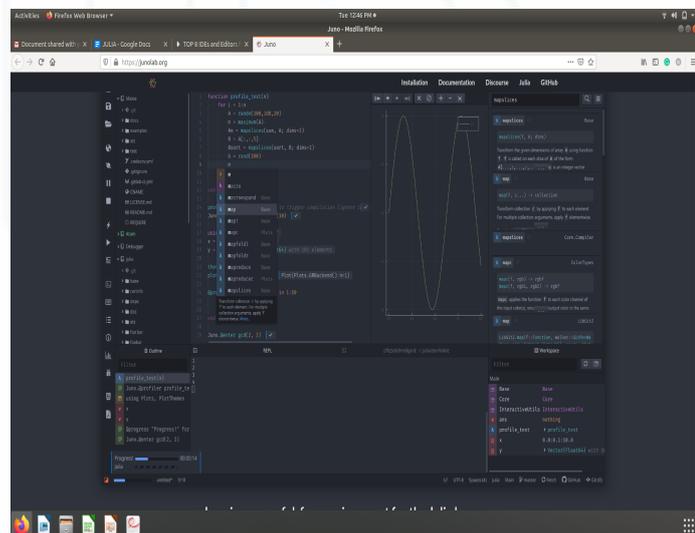
Terminal, Editors and IDEs

Different languages use different IDE(Integrated Development Environment) platforms to run the code efficiently. The use of an IDE is important because, firstly, it helps in appropriate debugging in the code. Secondly, it helps in efficient coding as it conducts proper unit testing. Lastly, it provides one with advanced features such as code profiling and refactoring.

Some IDE's and Editors for Julia Programming Language are discussed here:[5]

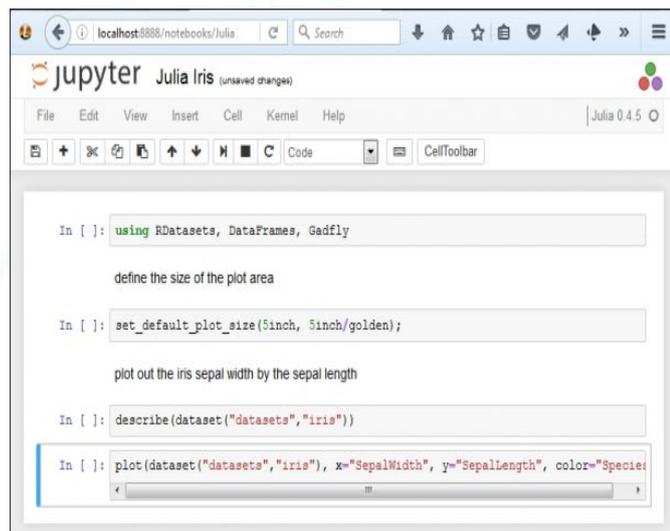
JUNO

Juno tops the list of IDEs for the Julia language. It is one of the most convenient and widely used IDE for this language. It is a platform that is equally comfortable for both beginners and developers. Juno offers Atom packages along with Julia so as to provide specific enhancements for Julia. It provides options like highlighting the syntax, plotting panes, running codes on console and a debugger for error correction. It also guarantees to provide a faster and more powerful environment for coding programs. Also, it makes it fun due to its canvas features. The way it helps you to customize your code using various advanced features such as using more than one cursor option is exceptional. This is how workspace in Juno looks like, convenient and customized.



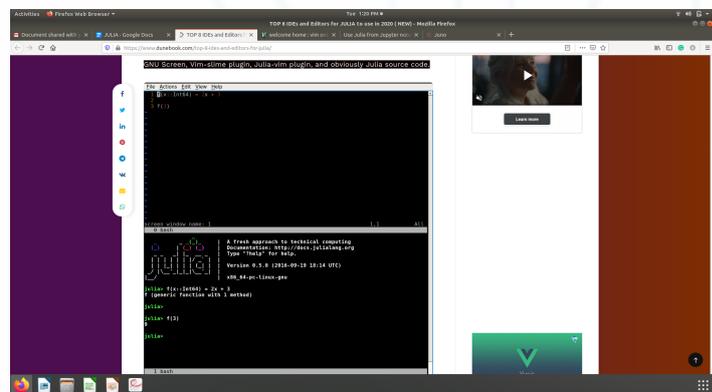
JUPYTER NOTEBOOKS

Jupyter notebook is another highly recommended IDE for Julia. It serves as an inter-connected platform. Apparently, it supports more than 30 different programming languages. Major tasks that Jupyter gives you access to are cleaning up of data and data transformation. Also, numerical solving and responses, data visualization, web coding, machine learning, statistical development and much more are provided. Other major advantages include that its notebooks can easily be shared. The output generated is really creative. Also, it has a centralized deployment and authenticity. It can manage large data really well. The Jupyter Notebooks have white background workspace that runs code quickly.



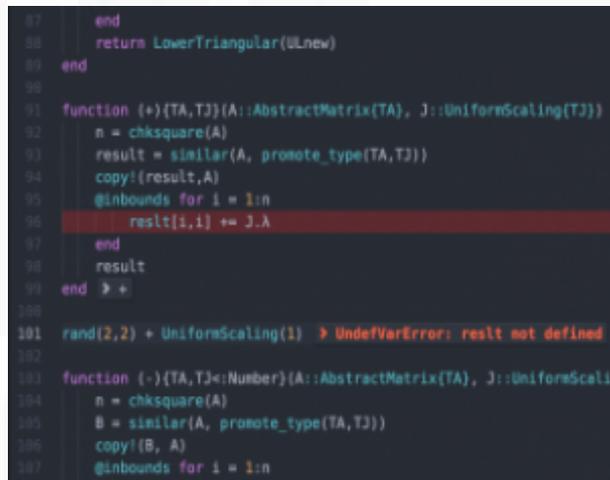
VIM

The third one on the list is Vim for its highly advanced customization features. It is one of the most versatile and extremely light-weighted editors. It works favorably for both beginners as well as developers. Also, it has got a whole lot of packages and tools for Julia language which gives it an extra point. The brief requirement list for Julia with Vim involves Vim editor, GNU Screen, Vim-slime plugin, Julia-vim plugin, and obviously Julia source code.



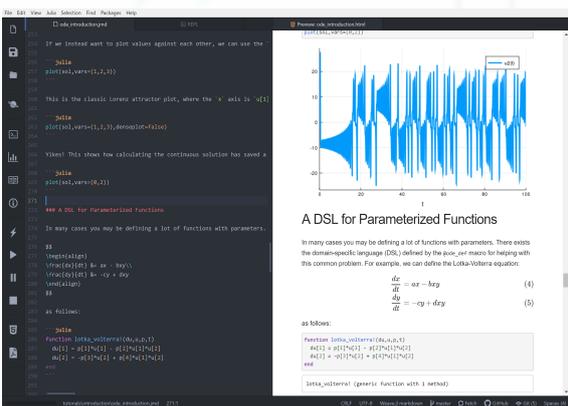
JULIAPRO

It is one of the most preferred IDE for Julia languages. It provides the user with high functionality to develop productive code. It is also responsible for providing boosted security, authentication, better and easy debugging and more efficiently for the same work. However, even developers can rely on this one for better environment purposes. Coding in JuliaPro looks like this. The initial versions of Julia had an issue with the size of it. The large files were not only difficult to download but they were also prone to error. Fortunately, the newer version (based on Julia 1.0) has solved this issue. It is quite compatible and has the only compiler and standard libraries



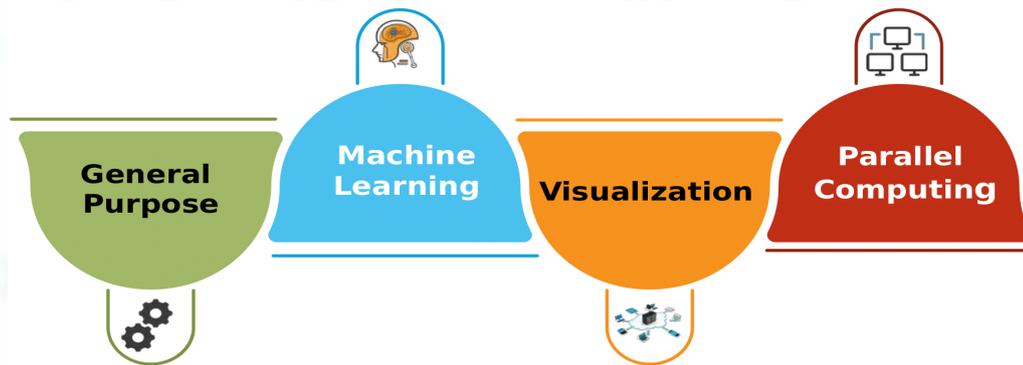
WEAVE

If you are thinking of using Julia for literate programming or data sciences, your go-to IDE is Weave. It is majorly used in creating data and scientific reports for programmers. Weave gives you an edge over the other IDEs in cases of mathematical programming. It also supports different formats for outputs such as GitHub, LaTeX and restructured outputs. The weave.jl package in weave provides extended features. These are scripting of documents for input and publishing the inputs as markdowns using Julia. It also helps in executing code in the form of 'terminals' or scripts, cache the output and convert the Julia documents.



Julia Ecosystem

The Julia ecosystem evolved around various area, some of them are explained below:



1. General Purpose Programming Language

Julia is a good general purpose programming language because of six main points.

1. It is light and efficient. There are use cases where very lightweight computers, such as a tiny computer, have been used with Julia to perform computationally intensive operations in real-time.
2. It is scalable. It doesn't need a big data framework in order to scale up. It's hard to overestimate the value of this advantage, especially in today's large-scale data world.
3. Julia is also a multiple-dispatch language. This allows it to have more intuitive function names that can handle multiple tasks, depending on the inputs given by the user.
4. Julia integrates well with existing programming languages that are popular among developers and data scientists. For example, you can call C, Python, and R scripts through Julia. This makes it easier to integrate Julia in work-flow, facilitating the whole data science pipeline.
5. Julia is a high-level language making it easier to understand and to manage Julia scripts. This ties very well with another advantage of the language, which involves its ease of learning.

2. Machine Learning in Julia

The resources for Machine Learning in Julia are still relatively distributed over different packages. Julia is not (yet) as popular as other programming languages for Machine Learning. The user needs to make more efforts to find (or write) certain basic data preparation functions that are easily available in Python and R. The good news is that there are initiatives to regroup Machine Learning models in larger libraries.

As a result, for some models, they simply provide wrappers to other, much smaller, Machine Learning libraries. Because of this, it can be important to cover two of those smaller libraries: "GLM" for Generalized Linear Models and "DecisionTree" for many tree-based models.

Some of the advantages of Julia in Machine Learning:

Speed and compilation:

The main advantage of Julia over other Machine Learning languages is speed. Two main reasons for Julia's speed advantage are, firstly, that it is a compiled language. Secondly, it has been designed for parallelism.

Script-like Syntax:

Despite the speed advantage, Julia's syntax is not very complicated. It feels much like a scripting language and it is relatively easy to switch to Julia from a Python or R background.

Community, developments, and libraries:

Julia is gaining traction and it is becoming more and more mature. With this growth, the community is also growing. Although still far smaller than the Python community, it is a good sign to see growth in both community and the number of libraries available.

Run Julia in Jupyter Notebook: Another great thing about Julia is that it can run in Jupyter notebooks. And it's very easy to install.

3. Visualisation in Julia

Data visualization is the process of representing the available data diagrammatically. There are packages that can be installed to visualize the data in languages like Python and Julia. Some of the reasons that make the visualization of data important are listed below:

- Larger data can be analyzed easily.
- Trends and patterns can be discovered.
- Interpretations can be done from the visualizations easily

The package that is used widely with Julia is Plots.jl. However, it is a meta-package that can be used for plotting. This package interprets the commands given and plots are generated using some other libraries and these libraries are referred to as backend.

The backend libraries available in Julia are : Plotly/PlotlyJS, PyPlot, PGFPlotsX, UnicodePlots, InspectDR, HDF5.

4. Parallel Computing in Julia

Most modern computers possess more than one CPU, and several computers can be combined together in a cluster. Harnessing the power of these multiple CPUs allows many computations to be completed more quickly. There are two major factors that influence performance: the speed of the CPUs themselves, and the speed of their access to memory. In a cluster, it's fairly obvious that a given CPU will have fastest access to the RAM within the same computer (node). Perhaps more surprisingly, similar issues are relevant on a typical multi core laptop, due to differences in the speed of main memory and the cache. Consequently, a good multiprocessing environment should allow control over the "ownership" of a chunk of memory by a particular CPU. Julia provides a multiprocessing environment based on message passing to allow programs to run on multiple processes in separate memory domains at once.

Parallel programming in Julia is built on two primitives: remote references and remote calls. A remote reference is an object that can be used from any process to refer to an object stored on a particular process. A remote call is a request by one process to call a certain function on certain arguments on another (possibly the same) process.

Sakshi Shah, FYIMSc-IT

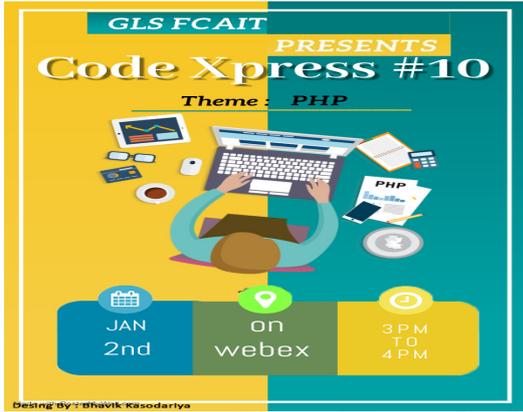
Conclusion

Other than mentioned above features, Julia also provides asynchronous I/O, meta-programming, debugging, logging, profiling, a package manager, and many more. People can build entire Applications and Micro-services in Julia hence now a days it is among top most languages uses over the globe for high-level dynamic programming language.

References

- [1] <https://www.simplilearn.com/things-to-know-about-julia-programming-language-article>
- [2] <https://www.zdnet.com/article/programming-languages-julia-users-most-likely-to-defect-to-python-for-data-science/>
- [3] https://en.wikipedia.org/wiki/Comparison_of_programming_languages#General_comparison
- [4] <https://docs.julialang.org/en/v1/>
- [5] <https://www.dunebook.com/top-8-ides-and-editors-for-julia/>
- [6] <https://www.forbes.com/sites/suparnadutt/2017/09/20/this-startup-created-a-new-programming-language-now-used-by-the-worlds-biggest-companies/?sh=b4558757de2a>
- [7] <https://www.analyticsvidhya.com/blog/2017/10/comprehensive-tutorial-learn-data-science-julia-from-scratch/>
- [8] <https://towardsdatascience.com/deep-learning-with-julia-flux-jl-story-7544c99728ca>

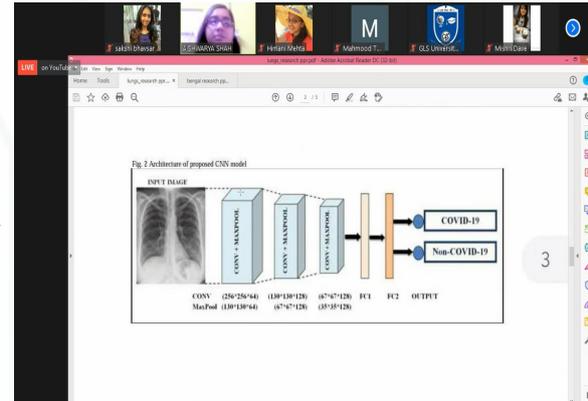
Code Express #10



Due to ongoing pandemic conditions, this year Code Express was organised online through the Webex platform on 2nd January, 2021 on the theme of PHP to develop coding skills across students. Students showcase their expertise in a friendly and spirited environment to develop their coding skills. This event inspired students to work on Web Development including Creating Webpages, Deployment and use concepts of Encryption and Decryption, Session Handling and Database Connectivity.

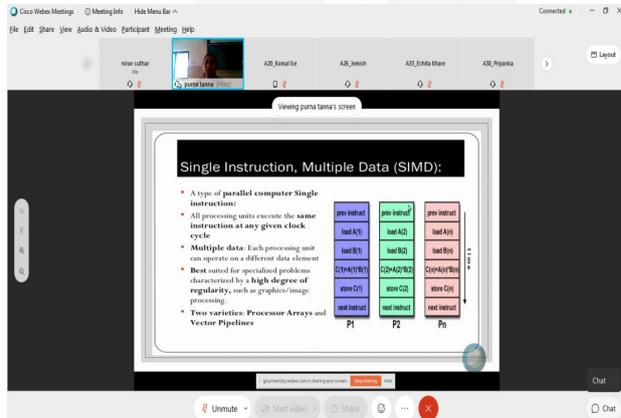
Tech Talk

Tech-Talks are technology-oriented talks that help students stay up-to-date in the usage of technology and become better communicators in the workplace. In this academic session, a Virtual Techtalk "AI v/s PANDEMIC" was organised on 25th February, 2021 and was live on YouTube. The talk was focused on common aspects of Artificial Intelligence and Machine Learning.



Webinar on Parallel Processing

To go beyond classroom teaching and curriculum, the institute has organized a Webinar on "Image Processing" on 6th January, 2021 by Dr. Purna Tanna, Assistant Professor, FCAIT, GLS University.



The session enlightened the students about the working of Parallel Processing, various applications and objectives of Parallel Processing.

It also explained different types of parallel techniques which enable the system to achieve simultaneous data-processing tasks to increase the computational speed of a computer system and how a parallel processing the system can carry out simultaneous data-processing to achieve faster execution time. The webinar was informative and helpful to explore the current market scenario in the market.

Alumni Talk

The institute is keen on being in touch with its alumni across the world. Keeping this in mind the Institute arranges sessions for Student-Alumni interaction. It provides students an exposure to alumni's work profile and culture. Moreover, students acquire knowledge about the recent IT trends and technologies.

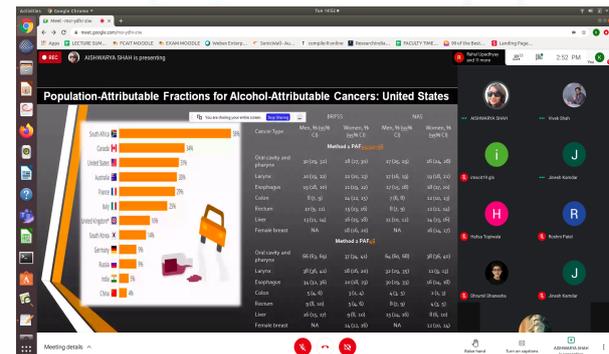
This semester we got connected to one of our alumni Ms. Dhara Gohel, Cyber Security Analyst, Wipro Technologies, Pune, India. She explained about different types of securities like E-mail, SOC & SEIM, File Integrity Monitoring etc. She also focused on how to manage the customer interface through arranging regular progress implementation meetings and ensuring project deadlines and involving in high severity issues P1/P2, taking remediation action without impacting cyber security.



Academic Activities

Hackathon 2020-21

INNOV INDIA HACKATHON: The Innov India Hackathon was organized by Innovation and Yuva Vertical, Confederation of Indian Industry on 31st July, 2020. Total 3 teams having 12 students participated in this competition from FCAIT having different projects such as CODE BLUE-SAVE LIVES (Patient monitoring system), Application for Migrant Worker and Face Recognition Attendance System. The team CODE BLUE won the 1st Prize and were offered an internship with a stipend of Rs. 6000 per month at Solvios Technology, ASAYA Infosolutions, Ahmedabad.



E-HACKATHON, SSIP, INDUS UNIVERSITY: The 10-hours virtual E-Hackathon was organized by Indus University, Ahmedabad, under SSIP, Gujarat Govt on 19th January, 2021. Total 4 teams having 24 students participated in this competition from FCAIT, with projects such as Smart Vehicle Sensor-2021, Being Abstemious-helping people to leave alcohol addiction, The helping hand-platform for the poor or needy people to get food, clothes and connect NGO.

IDEATHON, GEC RAJKOT: The Ideathon was organized by Government Engineering College, Rajkot- Under SSIP, Gujarat Govt on 22nd to 31st January, 2021. Total 2 teams having 12 students participated in this competition from FCAIT with ideas such as Being Abstemious - Helping people to leave alcohol addiction and GOLDEN EAGLES- Help people to complete their day with a fresh and good meal at a very minor cost.

IDEATHON - NATIONAL LEVEL SYMPOSIUM, ATMIYA UNIVERSITY, RAJKOT: The Ideathon was organized by Atmiya University, Rajkot - National Level Symposium on 27th February, 2021. Total 2 teams having 6 students participated in this competition from FCAIT with different projects such as being abstemious - Helping people to leave alcohol addiction and Online Crime Reporting System. The project "Helping people to leave alcohol addiction" secured 1st rank and "Online Crime Reporting System" secured 2nd rank.

MINED HACKATHON, Nirma University, Ahmedabad-BINGHAMTON UNIVERSITY, NEW YORK: Three days virtual hackathon was organized by Nirma University, Ahmedabad - BINGHAMTON UNIVERSITY, NEW YORK on 26th to 28th February, 2021. Total 2 teams having 9 students participated in this competition from FCAIT with projects entitled - Automated Robotic Arm and GOLDEN EAGLES - Help people to complete their day with a fresh and good meal at a very minor cost.

NMIT, NATIONAL LEVEL HACKATHON BY NIITE MEENAKSHI INSTITUTE OF TECHNOLOGY, BANGLORE - This Virtual Hackathon was organized by NIITE MEENAKSHI INSTITUTE OF TECHNOLOGY, BANGLORE on 20th & 21st March, 2021. Total 3 teams having 10 students participated in this competition from FCAIT with new ideas entitled - Smart Mirror, Anti-Cheat Portal and being abstemious - Helping people to leave alcohol addiction. The project Smart Mirror won with the 3rd rank.

PROTECH- 21, SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE: The 30 hours virtual hackathon was organized by Symbiosis Institute on 7th & 8th May, 2021. Total 3 teams had participated in this competition from FCAIT with projects entitled - Being abstemious - Helping people to leave alcohol addiction (IDEA PRESENTATION), Code Blue- Patient Monitoring System (IDEA PRESENTATION), Time Series Prediction of COVID-19 cases in GUJARAT (POSTER PRESENTATION). The project "Code Blue- Patient Monitoring System" won 1st rank with Protech 2021 momento and cash prize of Rs. 1000. and "Time Series Prediction of COVID-19 cases in GUJARAT" won 1st rank with Protech 2021 momento and cash prize of Rs. 2000. Also, "Being Abstemious - Helping people to leave alcohol addiction" won 2nd rank.

Academic Activities

CYBER SHADEZ- 2020-21

“Opportunities don’t happen, you create them”, adhering these words, Faculty of Computer Applications & IT (Msc-IT, BCA, PGDCA, iMsc-IT) and Faculty of Computer Technology (MCA) jointly hosted the 15th edition of a three-day International Webinar and Techfest “Cyber Shadez” with the theme of “INDUSTRY 4.0” from 11th to 13th February 2021. Embracing the new normal, the programme was conducted online.

The objective of the TechFest was to provide a platform to young students to gain an insight into Industry 4.0 which is the current trend of automation and data exchange in manufacturing technologies. Under Cyber Shadez 2021, "Importance of Data in Industry 4.0" and "Industry 4.0 in Financial Service", International Webinars on Industry 4.0 and various competitions like Tickle Your Mind (Idea Presentation), Bug Spotter (Bug Finding in WebSite), Code Jigsaw (jumbled program snippets in 'C'- Language), Python Charmer (Code Debugging), IKnowDB- Database Treasure Hunt, Brain-O-Brain (IT Quiz), Database Treasure Hunt, Programming and Logo Design were organized for UG and PG students and competitions like Webholic, Braindigger and Tech Teaser were organized for higher secondary school students too. Students from more than 15 universities from states of Maharashtra, Rajasthan, Telangana, Gujarat and Madhya Pradesh actively participated in Cyber Shadez 2021. The event received an overwhelming response where more than 500 participants from around 40 institutes across India zestfully participated. Cyber Shadez 2021 was also supported by the Gujcost, Department of Science and Technology, Government of Gujarat.

A three-day Tech Fest started on 11th February 2021 with the Inauguration of School Events in which more than 60 higher-secondary school students of various schools of Ahmedabad, Bharuch and Udaipur participated in competitions like WEBAHOLIC (Web Designing), BRAINDIGGER (Programming), and TECH TEASER (IT Quiz).

On day two, Cyber Shadez 2021 comprised of two International Webinars and an event called Techathon on the theme of Industry 4.0. Two major attractions of the programme were the international webinars -"Importance of Data in Industry 4.0" by Mr. Kedar Golvelker, Performance Tuning Analyst, Bunnings Warehouse, Australia, and "Industry 4.0 in Financial Service" by Ms. Pooja Padgavkar Senior Consultant Smart Stream Technologies, UK. Mr. Kedar shared his views on SQL Tuning, Securing Data, tips and techniques to write SQL, use of Data Science and Data Analytics and Data Management in Industry 4.0 and Ms. Pooja gave a glance on design principles of Industry 4.0, impact and benefits of Industry 4.0 on the financial services sector, a role of AI in the field of financial services and pre and post-AI era.

Post-webinars the longest competition of the event – Techathon was organized for UG and PG students where 8 teams had participated with their innovative solutions for issues related to Covid-19, healthare and agriculture. Day 3 comprised of the various technical events like Tickle Your Mind, Bug Spotter, Code Jigsaw, Python Charmer, Database Treasure Hunt, Brain-O-Brain, Programming and Logo Design for UG and PG categories. The day was ended with a valedictory ceremony of Cyber Shadez 2021.

At the end, the result of the competitions were announced and prizes were awarded for UG and PG events.



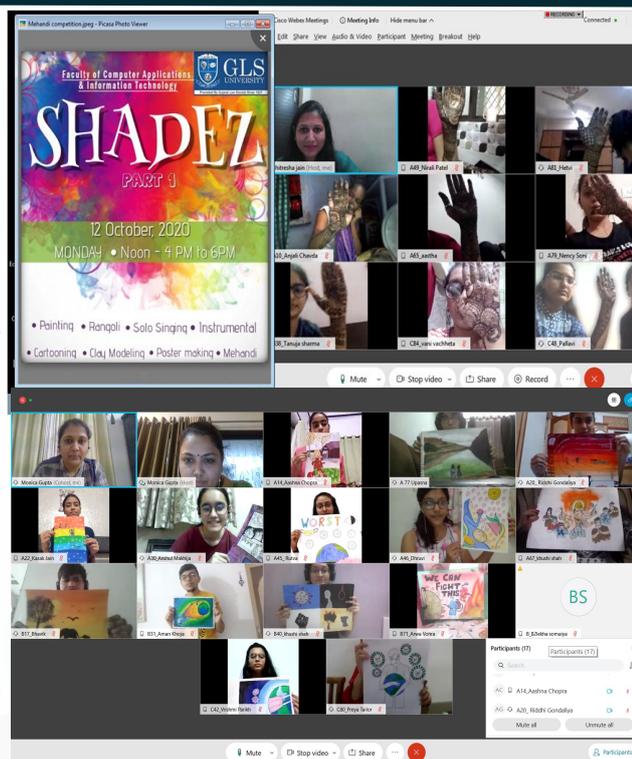
SHADEZ 2020-21

College cultural events are one of the most happening events which brings students together. Cultural activities create social solidarity and cohesion, fostering social inclusion, community empowerment and capacity-building, enhancing confidence, civic pride and tolerance.

It's a day where one event can make a lifetime memory. SHADEZ – 2020” is an annual cultural fest organized by FCAIT, GLS University. This year it was held on 12th October, 2020 and 9th January, 2021 in virtual mode.

On 12th October, 2020 various events such as singing, painting, rangoli, instrumental, poster making and mehendi were organized in which more than 70 students participated. Students showcased their talent and enjoyed throughout the event.

On 09th February, 2021, events like poetry writing, poetry recitation, elocution, solo classical dance, solo western/fusion dance and stand up comedy were organized.



Faculty Achievements

1. Ms. Tripti Dodiya, Dr. Ali Yawar Reha, “Using Term Frequency - Inverse Document Frequency to find the Relevance of Words in Gujarati Language”, Volume 9, Issue IV, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 378-381, ISSN : 2321-9653, April 2021.
2. Ms. Tripti Dodiya, Dr. Ali Yawar Reha, “Part of Speech Tagger Using HMM for Gujarati Language”, Volume 11, Issue 4, Journal of Information and Computational Science (JOICS) Page No. 199 - 208, ISSN: 1548-7741, April 2021.
3. Ms. Disha Shah, Assistant Professor, Faculty of Computer Applications & IT (FCAIT), GLS University was awarded the degree of Ph. D by Pacific University, Rajasthan under the guidance of Dr. Neetu Agarwal. The topic of her Ph.D. research project was “Building A Single Point Interface for Data Analysis of Data Gathered using Wireless Sensor Network”.
4. Jyoti R. Dubey and Ankit R. Bhavsar published a research paper titled “Simulation and Performance Evaluation of WSN Based Architecture for Fleet of Long Route Vehicles Using NS2”, High Technology Letters Journal, Volume 27, Issue 1, ISSN: 1006-6748, January 2021.
5. Prof. Hemali Moradiya and Dr. Kalpesh Popat presented a research paper titled "Intelligent Congestion Control System to Control Computer Traffic by using various techniques - A Survey" in the Journal of International Journal of Innovative Science, Engineering and Technology, Volume 8 , Issue 2, ISSN: 2348 - 7968, February 2021.
6. Prof. Vidhi Thakkar and Dr. Vrushank Shah presented a research paper titled "Investigation of Techniques Used For Mitigating Security And Privacy Issues In Cloud Based Electronic Health Record Systems" in the Journal of International Journal of Innovative Science, Engineering and Technology, Volume 8, Issue 2, ISSN: 2348 - 7968, February 2021.
7. Prof. Nirav Suthar and Dr. Ankit Bhavsar presented a research paper titled "WSN Based Prototype Architecture for Alerting the Effects of Forest Fire on Wild Animals in Gir Forest Gujarat" in the International Conference on Innovative Computing and Communication(ICICC-2021) organized by Shaheed Sukhdev College of Business Studies, University of Delhi, New Delhi, India in association with National Institute of Technology Patna, India and University of Valladolid, Spain on 20th-21st February, 2021.
8. Prof. Snehal Shukla and Dr. Purna Tanna, published a research paper titled “Pattern Recognition In Gujarati Character Detection– A Pragmatic Approach” in International Journal for Science and Advance Research In Technology, Volume 6, Issue 12, ISSN: 2395-1052, December 2020.
9. Prof. Rachna Chaudhari and Dr. Purna Tanna, published a research paper titled “Analysis of Handwritten Joint Characters in Gujarati Language” in International Journal for Research in Applied Science and Engineering Technology, Volume 8, ISSN: 2321-9653, December 2020.