A GUIDE TO PRODUCTIVE SUMMERS

Rishabh Jain

+91 97998 23983

f2013473@pilani.bits-pilani.ac.in

Anirudh Agarwal

+91 77288 35763

f2013644@pilani.bits-pilani.ac.in

Acknowledgements

The matter has been contributed by the following:

- Priyans
- Bittu
- Suraj
- Jaideep Singh
- Sundar Ganesh
- Mathew M John
- Ashish
- Shaswat Verma
- Ankur Baheti
- Ramchandran
- Kushagra Nigam
- Raj
- Palash
- Sajal
- Siddharth

Special thanks to Raj Kunkolienkar & Shilpi Agrawal.

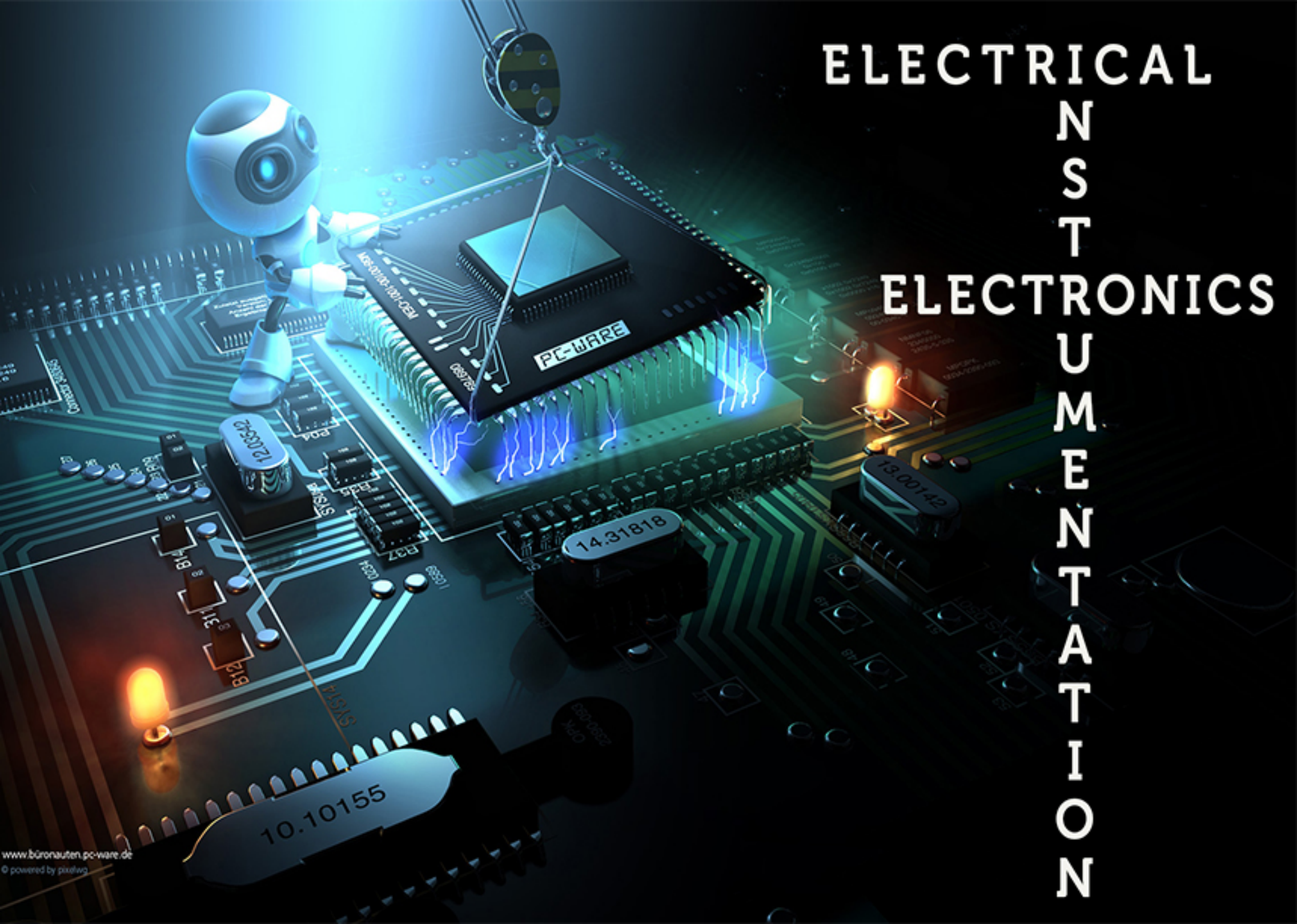
DISCLAIMER

This intention of this document is to help answer the question that has been known to plague people for quite some time now.

"What do I do after my first year on campus? How do I make the best out of the summer in hand?"

After A YEAR of foundation courses, one would typically develop enough of a base to get started with explorations in different areas of academic interest. The aim of this document is to pass down valuable resources to tools and field that one can jump into.

If you have anything to contribute, feel free to briefly jot down what you have in mind and drop a mail at f2013644@pilani.bits-pilani.ac.in.
Do mention the prerequisites and attach a link that'll better guide people.
You are welcome if you are willing to guide people on the same.


ELECTRICAL
INSTRUMENTATION
ELECTRONICS
PC-WARE
14.31818
13.00142
12.00542
10.10155
www.büronauten.pc-ware.de

Buy lots of hardware.

Start making circuits small circuits.

Buy Arduino boards, Solder wires, resistors capacitors.

Set up a small DIY-Lab.

Buy sensors, radios, play around experiment with it.

Make LEDs blink, make things move, do all sorts of cool stuff with it.

If you want to know where to buy hardware from :
http://goo.gl/t8Fgt3 .

For serious people who would want to set up a lab in their hostel rooms / homes - http://goo.gl/vag4A

Hack your normal electronic appliances, break them open, see how they work, understand, read and blog about it.

If you are the one who would not want to buy hardware or are not into this sort of thing, learn PCB designing.

Open-source tools like KDiCad are available, or you could start with Eagle too.

Look for interesting circuits to design, get them fabricated. Solder components on them and play away!

If you are still not into PCB designing or hacking circuits, experiment with Pspice (You must have already learnt it during ES1 course).

Start simulating bigger and bigger circuits. Move on to complicated circuits! Move to tools like Proteus and Cadence.

Learn VHDL, Verilog. These are proper languages which use programming to simulate hardware. Resources are all available online, get started!

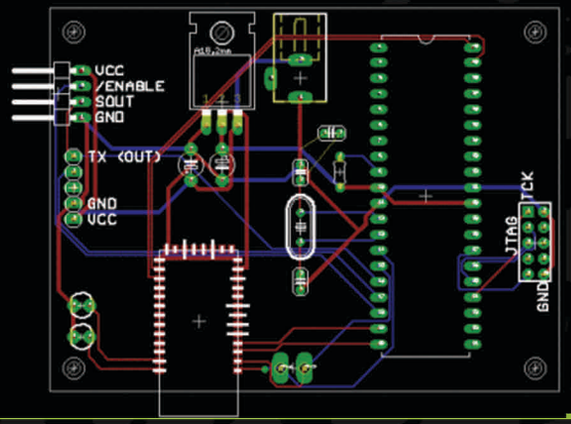
VHDL2VeriLog

You can build all sorts of cool stuff, like simulating hardware which is highly optimized to run image processing algorithms.

You need to appreciate the beauty of digital logic and circuitry to love this. Publish all your codes on GitHub!

Even if this does not inspire you to do something, read up latest research going on in fields like MEMS, Nanotech, Bionics, Material Sciences, Biosensor systems.

github
SOCIAL CODING



Bookmark hackaday.com and visit it daily.

Consider shifting to a linux distribution as your primary OS.

If you like learning from a book, have look at "The Art of Electronics" by Paul Horowitz

Buy an Arduino. If you cant afford it, buy a Freeduino. They're about half the price and electrically equivalent.

Learn how to solder. Solder well. Buy a soldering iron and an extra tip.

Learn how to use LTspice (or some equivalent). Can't stress enough on how helpful it is.

Checkout-http://ozark.hendrix.edu/~burch/logisim and http://www.logiccircuit.org/.

Move on to Proteus only when you are comfortable with the above.

Create a log of what you have learnt, even the mistakes. Preferably keep it online and up to date.

HACK A DAY

If you are really into embedded systems, don't use the arduino I/O libraries for too long. Move on to plain c.

You will learn a lot more from programming a 16 bit timer than you will ever learn from analogWrite().

Learn how to program a chip using avrdude and only then move onto an IDE or custom build scripts.

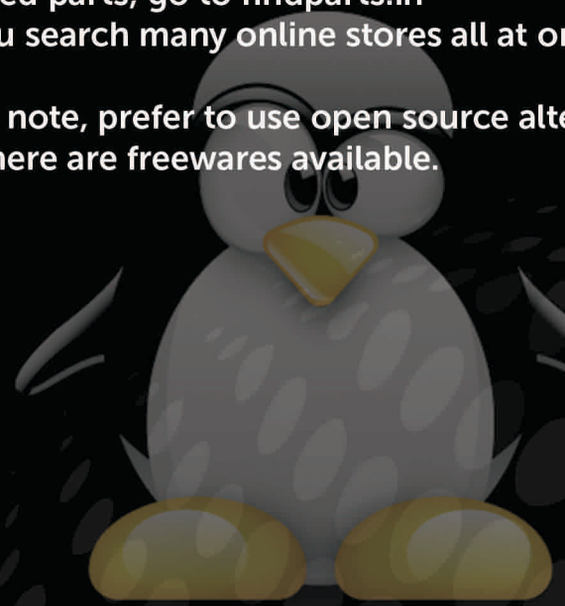
Bookmark http://avrfreaks.net and follow https://www.youtube.com/user/jeriellsworth/videos

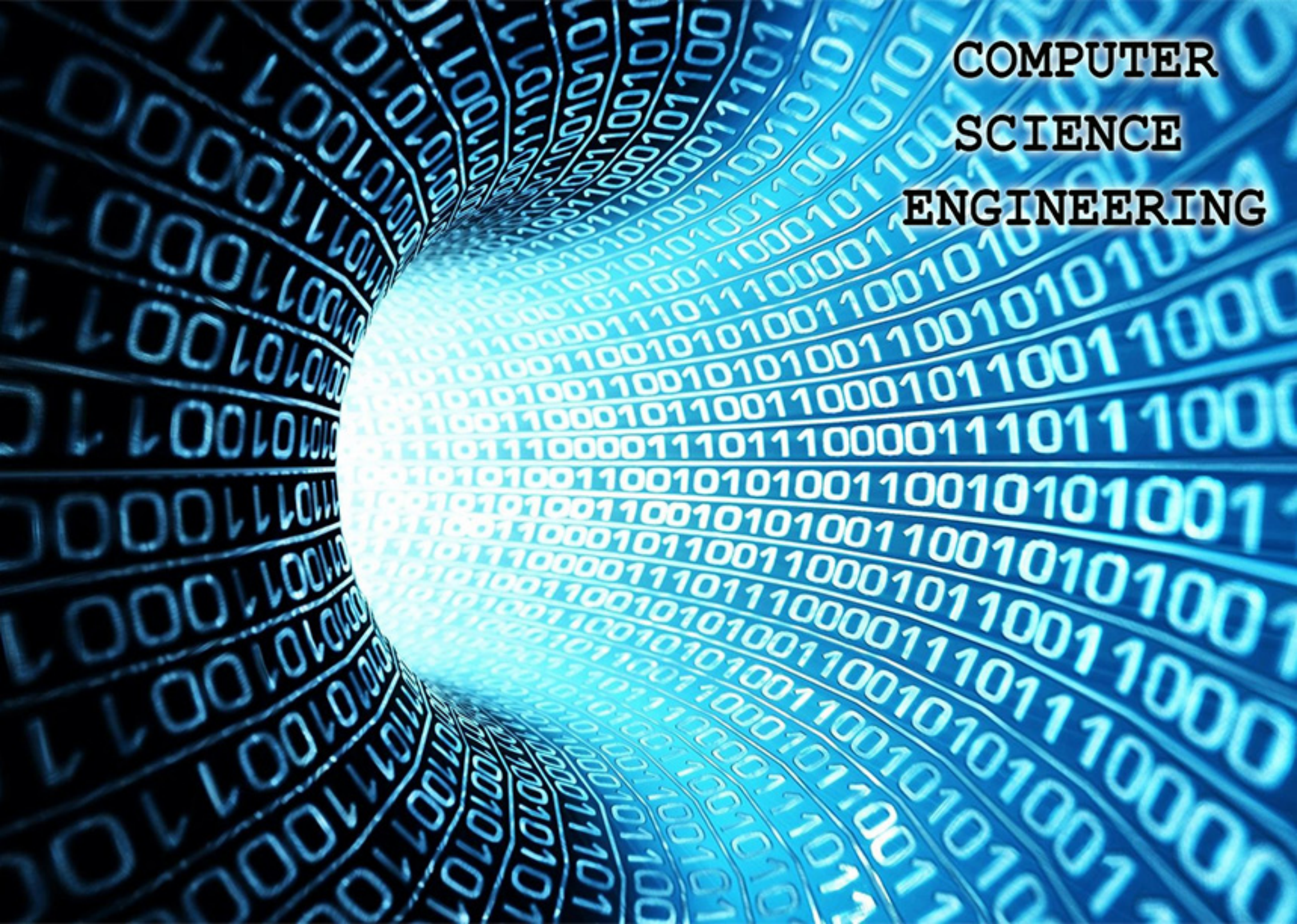
Learn how an IC 555 works. Better try to think about why its called that (hint: look at the internal block diagram).

Check out https://learn.sparkfun.com/tutorials/tags/concepts and try a few small projects to get started.

If you need parts, go to findparts.in
It lets you search many online stores all at once.

As a side note, prefer to use open source alternatives even if there are freewares available.




COMPUTER
SCIENCE
ENGINEERING

For **Beginners**

- For students who are new to programming practice your coding skills and master the basics of an object-oriented language like C++/Java/Python.
- Get a firm grasp on OOP concepts like polymorphism, inheritance, abstraction and basic data structures like stacks, queues, heaps and algorithms on sorting and searching along the way.
- I would recommend you to continue with C++ since it's similar to C which you already learnt. You would be anyways be learning Java as part of OOP in your next semester.
- Recommended tutorials: (for Python), http://www.learncpp.com/
  http://www.cplusplus.com/doc/tutorial/ (for C++)
  http://www.codecademy.com (for HTML, CSS, JAVASCRIPT, etc)

For **Advanced**

- Enter into the world of competitive programming problems along with explanations.Have a read -
  Check out www.spoj.com/ www.codechef.com/ www.topcoder.com/ www.hackerrank.com.
- Create an account and start honing your algorithmic skills. Most of these sites have very good tutorial.
- Try doing a MOOC on algorithms. It really helps. Don't wait for your DSA course in 4th sem.
  Coursera has 2 upcoming courses from Stanford
- Algorithms 1 - https://www.coursera.org/course/algo
  Algorithms 2 - https://www.coursera.org/course/algo2- You get a certificate of accomplishment as well :)

Open-source Software

- The best part about open-source software is its vibrancy and 24/7 active community ready to help you even from scratch.
- Have a look at https://www.google-melange.com/gsoc/org/list/public/google/gsoc2015 to see a list of popular open-source orgs. Filter out them according to your preferred fields and choose one(or more).
- Almost all these organizations have a documentation with information on building and hacking the software. Along with this, they also generally have an IRC or mailing list where other contributors hang out. Introduce yourself, tell them you are interested in contributing. You will find plenty of nice people to help you out.
- Also, do check out http://openhatch.org/
- If you are into OS kernels and want to know how operating systems work - http://eudyptula-challenge.org/ You build a kernel from scratch and add features on top of it.
- Get yourself a www.github.com account and learn to use git.
- If you would like to tackle really good problems in algorithms or you would like to improve your coding skills, take a look at Project Euler. It has challenging Mathematics and Computational Problems.
- Start working on simple projects after CP. Try and recreate basic utilities you have used.
- it's important to realise that open source development is a bit different from all the programs you've been writing so far. Practically, the code base is huge and managing it is an issue.
- The first thing is you need not be an amazing coder or awesome developer (who has all algorithms in his fingertips) to start contributing to Open Source. You just need to be willing to work hard.

Some Interesting Topics

1. Bit manipulation/Bit twiddling (http://graphics.stanford.edu/~seander/bithacks.html)
2. Virtual machines and operating systems.
3. Networks (via ethical hacking or penetration testing, if you wish to)
4. SSDs
5. Big Data, Parallel and Distributed computing.
6. Heartbleed, Man in the middle attacks and computer security in general.

Internet Relay Chat

One of the oldest ways to communicate over the internet. Begin with installing an IRC Client (XChat is cross-platform and has a decent interface). Most open source projects have channels (similar to chat-rooms) on the Freenode network (irc.freenode.net). You might want to configure the client to connect to Freenode automatically on launch. Also, select a nickname - this is going to be your identity over IRC. Having a funny nick is fine but don't go overboard with it. If you get a prompt asking for password, it implies that the nick you selected has already been registered and you'll have to pick another one. Once you've decided on a nick, register it so that no one hijacks it. Join your organisation's channel and ask questions right away (be polite, don't ask to ask). You'll be surprised to see how helpful the community is. Refer to http://xchat.org/docs/start/ for configuring XChat.

Open-source Software

- The best part about open-source software is its vibrancy and 24/7 active community ready to help you even from scratch.
- Have a look at https://www.google-melange.com/gsoc/org/list/public/google/gsoc2015 to see a list of popular open-source orgs. Filter out them according to your preferred fields and choose one(or more).
- Almost all these organizations have a documentation with information on building and hacking the software. Along with this, they also generally have an IRC or mailing list where other contributors hang out. Introduce yourself, tell them you are interested in contributing. You will find plenty of nice people to help you out.
- Also, do check out http://openhatch.org/
- If you are into OS kernels and want to know how operating systems work - http://eudyptula-challenge.org/ You build a kernel from scratch and add features on top of it.
- Get yourself a www.github.com account and learn to use git.
- If you would like to tackle really good problems in algorithms or you would like to improve your coding skills, take a look at Project Euler. It has challenging Mathematics and Computational Problems.
- Start working on simple projects after CP. Try and recreate basic utilities you have used.
- it's important to realise that open source development is a bit different from all the programs you've been writing so far. Practically, the code base is huge and managing it is an issue.
- The first thing is you need not be an amazing coder or awesome developer (who has all algorithms in his fingertips) to start contributing to Open Source. You just need to be willing to work hard.

More on Open Source

Be a bit familiar with the syntax, and basic concepts like functions, classes. And then have a look at various open source projects (preferably the ones that have a history of having taken part in Google Summer of Code, since they are the ones with rapid development). At first, the sheer complexity of the entire codebase might frighten you, this is normal even for an experienced coder, these projects have been worked upon by the best developers across the world for quite a bit of time. This is where the mailing list and IRC comes into play. Ask smart questions like which part of the codebase you should be working on, they would most probably lead you to a number of issues/bugs that need to be fixed.

If these issues seem a bit complex at first sight, relax and start working on it slowly. For a first timer, these issues would take at-least a minimum of 2-3 days to fix. There are always sites like stackoverflow.com that would help you if you get stuck anywhere and of-course the mentors of the project who are also willing to help. Once you are done, push your code, to the repository, and they would review it.

Now, reviewing also takes quite a bit of time, since the mentors work on these projects only in their free time. After the complete cycle of reviewing and rewriting code again is over, they will finally merge it and you would have your first Open Source contribution.

Version Control

With many individuals contributing to the project simultaneously, keeping “revisions” of files becomes a necessity. Consider this scenario - you figure out a bug that can be resolved by making changes to function foo in xyz.py. Now, what if the file xyz.py was updated during the period you last fetched it and applied your changes? How do you keep a track of who updated the code? This is where version control comes in. It helps you keep a track on every change made to the project and maintains “revisions” of files in the project. There are two types of VCS - Centralised (SVN, CVS) and Decentralised (git, hg).
I’d suggest you to get familiar with the one that the organisation of your choice uses (although being familiar with git is an added advantage due to its popularity). Nettuts has an introductory tutorial for using git **http://code.tutsplus.com/tutorials/easy-version-control-with-git--net-7449** .
Also refer to **http://git-scm.com/book**

More on Competitive Programming

Go through the IIT KGP C_Programming videos on DC which is an extension of our Computer Programming course. It contains a few concepts on algorithms (design and analysis for new-timers) and also introduction to graphs and trees.
When you are done with Linked Lists practice the problems on this website :**http://www.geeksforgeeks.org/-category/linked-list/.**
www.codeforces.com conducts 2 hour contests almost every 3-4 days. It contains 5 problems of good standard. Codechef also conducts regular contests. But do remember to go through the contests conducted by various colleges like IITs and IIITs in contests section of www.codechef.com . If you feel you are comfortable and strong with the basics of programming , then i would suggest you to go through basics of algorithms, sorting and searching in this website **http://www.geeksforgeeks.org/fundamentals-of-algorithms/** . It contains the source code to few algorithms which may help you to understand the algorithm better.
For data structures refer : **http://www.geeksforgeeks.org/data-structures/** .
Note: Participation in contests is very important. Even though you know the logic, its implementation is very important.

Mechanical Engineering

AeroSpace Engineering

Primary focus on Computational Fluid Dynamics (CFD) for interested since BITS does not separately offer an Aerospace Engg. program.
Broadly speaking, Aerospace Engg. has four main subdivisions - Aerodynamics, Structures and Materials, Control Systems and Propulsion. Being a Mechanical or Chemical engineering student, most of John D. Anderson's books should be simple enough to read and will give you a practical flavour of what you will encounter in Aerospace Engineering. I would strongly recommend a a First Year to start with this.

Mechanical Engineering student should pay attention to the following CDCs - Fluid Mechanics, Applied Thermodynamics, Mechanics of Solids, IC Engines, Heat Transfer, Prime movers and Fluid Machines and Gas Dynamics. AIs a Physics student special attention Computational Physics, Statistical Mechanics and Classical Mechanics. These should give you an extremely strong foundation of the physics involved in most Aerospace Engg. subjects. The prescribed textbooks for these courses are also extremely detailed, rigorous and simple to understand.

CFD is the use of a computer and various algorithms to simulate fluid flows around objects such as aircraft, turbines, jets, etc. This helps to obtain valuable data without the need to actually set up an experiment. You will be using programs such as ANSYS Fluent, CFX, STAR-CCM or OpenFOAM for these simulations. These programs use a GUI to make life much easier, but do not reveal any of the algorithms that they use and more often than not, don't allow the user to affect the algorithm used.

A First Year student can begin with tutorials for Fluent and OpenFOAM that can be found at these links-
https://confluence.cornell.edu/display/SIMULATION/FLUENT+Learning+Modules

http://www.openfoam.org/docs/user/index.php

OpenFOAM is an Open Source project based on C++ and online support through forums, libraries, tutorials and documentations is widely available. Starting with Fluent of OpenFOAM gives you very easy output with very little input and is extremely gratifying for a beginner to experience.

You can follow Youtube videos of:
1. Prof. Srinivas Jayanthi - IIT Madras
2. Prof. Suman Chakraborty - IIT Kharagpur

Learning numerical methods and computational programming is a must. Familiarize yourself with Matlab, Python, C, C++, FORTRAN or any other language specifically aimed at computational programming. Computational programming and numerical methods is a separate subject in itself. Follow these links:
1.)http://lorenabarba.com/blog/cfd-python-12-steps-to-navier-stokes/
2.)https://www.coursera.org/course/scicomp
3.)http://www.nptelvideos.in/2012/11/numerical-methods-and-programing.html
4.)https://www.edx.org/course/v2/introduction-computational-thinking-data-mitx-6-00-2x-0

I would also VERY highly recommend reading about the algorithms behind CFD as this is where the actual science and research happens. There are several textbooks you can use to get started - listed in the order that I would recommend reading them:
1. Introduction to Computational Fluid Dynamics - Versteeg and Malalasekara
2. Computational Fluid Dynamics - Basics with Applications - John Anderson
3. Computational Methods for Fluid Dynamics - Ferziger
4. Numerical Heat Transfer and Fluid Flow - Suhas Patankar

AUTOMOTIVES

Mechanical is one of the most diverse fields of engineering; Make a decision on which of the many skills you'd like to acquire over the next three years? (starting from your 1st Year summer vacations).

The answer to the above question majorly lies in your area of interest. For instance, we mechanical engineers have an option to work in:-

- Power plants - as field engineers, design engineers for various related systems.
- Automotive industry - Design engineers, CAD/CAE personnel, manufacturing floor
- Product design Industry
- Industrial Design Engineer(New in India)….. and the list goes on.

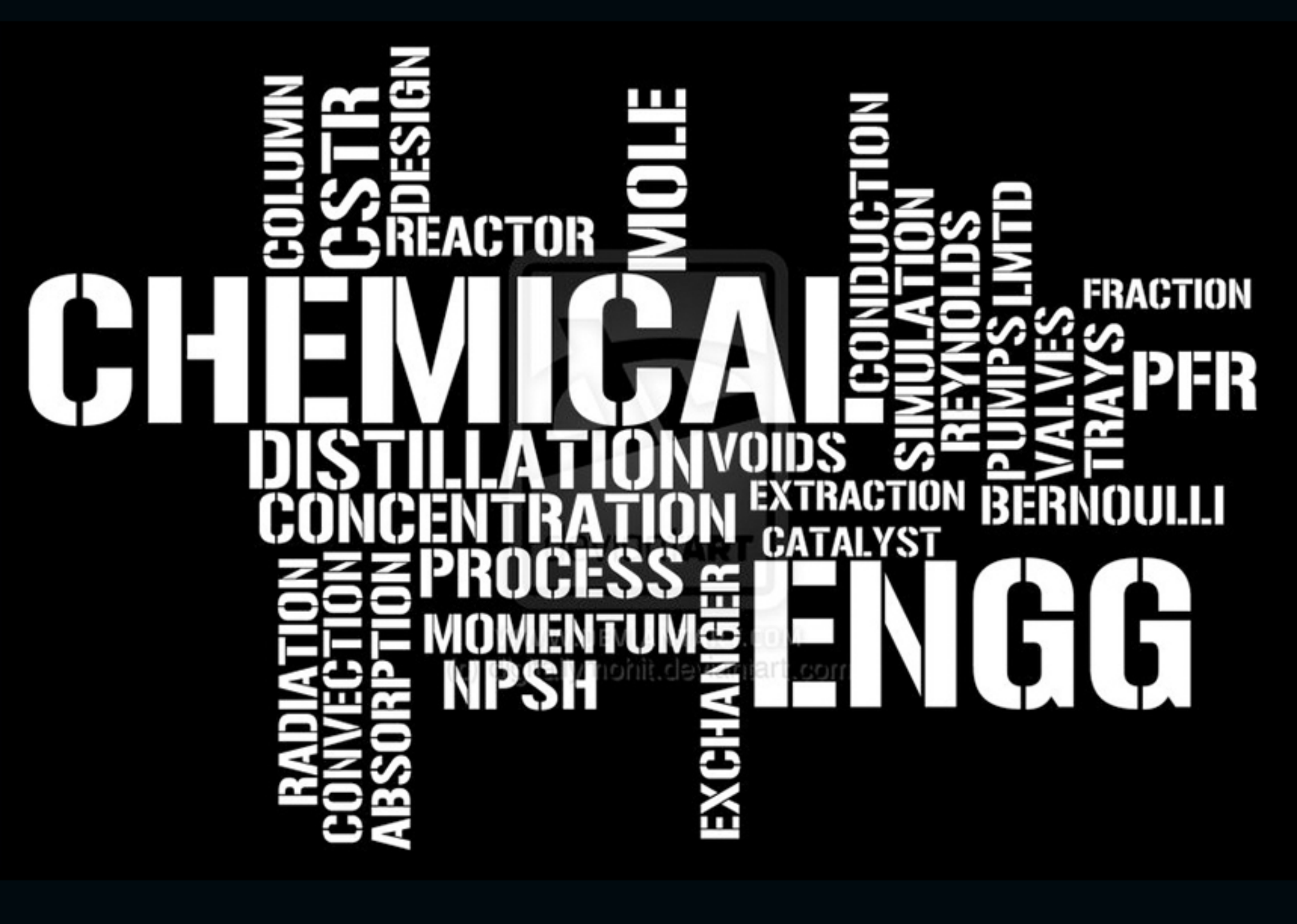
Once you've picked your area of interest, start browsing the web for additional info related to your field of interest. You'll find many forums, online courses, websites and e-books which will help you in your journey. Make a wise decision on what topics to read and what to ignore at this primary stage. It's really hard for anyone to guide you on all the areas related to Mechanical Engineering.

Engines, vehicle dynamics, vehicle construction, Chassis design, manufacturing process, CAD/CAE stuff and a lot many things go around in building an automobile.

If you are among those who had a knack for Engineering Graphics (AutoCAD) course in your 1st year then you can continue that momentum by learning new modelling softwares like:
-SolidWorks(Easy to work with and beautifully laid out)
-Pro Engineer
-CATIA V5 or higher version(One of the most advanced packages)
Start off modelling simple shapes and try to explore different options in the workbench, you can then try to make assemblies from a set of components you made. important to note that it's not necessary for you to master the software; You don't need to master any package yet but its always helpful to know them.

If you love reading books and are a frequent visitor on HowstuffWorks.com then you should probably keep reading books on Engines, vehicle dynamics or any other thing which you like., the aim here is to explore as always, get in depth knowledge on a specific topic.

Few good books which i'd suggest are:
-Internal combustion Engines by V Ganesan(best book for beginners)
-Internal Combustion engine fundamentals by John Heywood(a bit advanced and useful for MS years)
-Fundamentals of Vehicle Dynamics by Thomas D.Gillespie(neatly laid out book-medium)
-Advanced vehicle technology by Heinz Heisler(Simple with many illustrations, helpful)
In addition to all these(get your hands on one book at a time and keep browsing, understand how much ever you can) you can keep looking for online resources to keep you interested in new technologies etc.
-Formula1.com/news/technical (for formula1 fans)
-Howstuffworks.com
-carbibles.com
-and there is always youtube.com
Note: These online links are only useful to get you started and help build your intuition, this is how you'll always know something -in the back of your head- when someone speaks out a technical word.
Most important, enjoy doing whatever you are doing. You can start implementing whatever you learn during the next few years at BITS through college events or other events.


COLUMN
CSTR
DESIGN
REACTOR
MOLE
CONDUCTION
SIMULATION
REYNOLDS
PUMPS LMTD
VALVES
TRAYS
FRACTION
PFR
CHEMICAL
DISTILLATION
VOIDS
EXTRACTION
BERNOULLI
CONCENTRATION
CATALYST
PROCESS
ENGG
RADIATION
CONVECTION
ABSORPTION
MOMENTUM
NPSH
EXCHANGER

You will have four CDC's in your second year. Fluid dynamics, Engineering Chemistry, Thermodynamics and Chemical Process Calculations. You can read up and acquaint yourself with these. Thermodynamics and Fluid are possibly the more important ones and slightly harder too. You will need them when you come to your 3rd year and even later on. While reading up on these topics, also try to look into how and where they are applied in our day to day life. For example, thermodynamics plays a big role in the working of the refrigerator you have in your house. Furthermore, for chemical calculations, it helps to know a coding language. At BITS, MATLAB is the preferred choice in the second year. So try to equip yourself with basic knowledge in that. Your first year thermodynamics and chemistry courses will help you too, so brush up on that.

If you can get your hands on the prescribed textbooks beforehand, they're pretty good. Else you can read up on the internet or refer other books.

Prescribed textbooks:

Introduction to Fluid Mechanics-Fox, Pritchard and Mcdonald

Introduction to Chemical Engineering Thermodynamics-Smith and Abbott

Engineering Chemistry-S.Vairam, P.Kalyani and Suba Ramesh

You can go through some of the material given below.

http://ocw.mit.edu/courses/chemical-engineering/10-40-chemical-engineering-thermodynamics-fall-2003/readings/

http://nptel.ac.in/courses/103101004/1

If you already have an area of interest, work on that. You can also consider doing a small project or paper in that field which you can present at Quark and other techfests. Once on campus, the ME students will assist you with it.

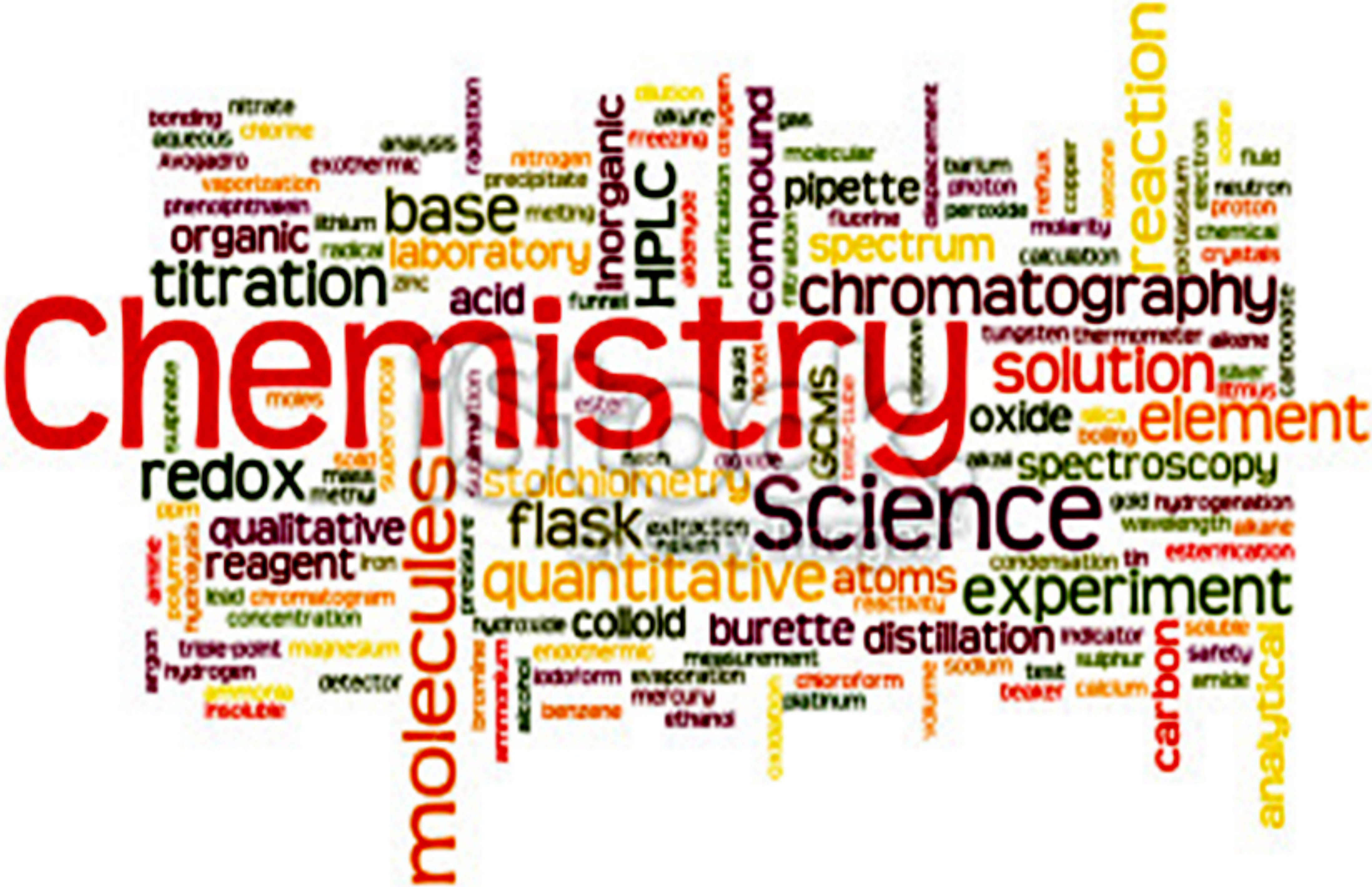

What can I do with Biology?
Pharmacology
Cell Biology
Environmental Science
Dentistry
Ecology
Biotechnology
Life Sciences
Nursing
Physiology
Conservation
Biogeography
Forensic Biology
Physiotherapy
Microbiology
Teaching
Equine Science
Farming
Midwifery
Molecular Biology
Medicine
Veterinary Science
Genetics
Parasitology
Zoology
Anatomy
Biomedical Sciences
Sports Science
Human Biology
Biochemistry
Neuroscience
Pathology
Psychology
Marine Biology
Biology
Botany

If you are into the pure biology, read up on techniques like PCR, ELISA, Gel electrophoresis etc. You will eventually learn about them. But they are really fun to use and very important. If you are looking for something interdisciplinary, Google Biology + <whatever your second interest is> There are lot of forums and videos on YouTube that explain the fields in good detail. It's a good starting point to learn new thing.

While core biology is still predominantly a lab oriented subject, there are many things you can do just sitting at home. One of the most fundamental things is coding. There is a fundamental shift in how biology is done off late. Given the vast amount of data generated, people are looking for ways to classify and analyze this data better. Start off by using software such as Cytoscape and PyMOL. While these are already rich in functionality, you can start creating your own plugins (Cytoscape works with Java and PyMOL works with Python).

The second aspect is getting used to research papers. Books can only take you so far in Biology. Concepts change on an almost-everyday basis and new things are discovered in biology at a rapid pace. Use PUBMED - this allows you access to a large number of research papers free of cost. Start reading and ask questions.

Finally, get working with a lab as soon as you can. You need lab experience to be good with Biology, theoretical knowledge will not cut it. Get in touch with a professor and request him/her to put you under a PhD student who can guide you.


Chemistry
chromatography
science
reaction
titration
base
organic
laboratory
inorganic
HPLC
compound
pipette
spectrum
acid
solution
element
oxide
spectroscopy
redox
molecules
stoichiometry
flask
qualitative
reagent
quantitative
atoms
experiment
colloid
burette
distillation
carbon
analytical
GCMS
nitrate
bonding
aqueous
chlorine
analysis
Avogadro
exothermic
nitrogen
precipitate
vaporization
phenolphthalein
lithium
melting
radical
zinc
funnel
freezing
alkyne
dilution
gas
molecular
barium
photon
peroxide
fluorine
molarity
calculation
neutron
proton
chemical
crystals
fluid
tungsten
thermometer
alkane
silver
litmus
silica
boiling
moles
ester
gold
hydrogenation
wavelength
alkene
esterification
condensation
tin
extraction
iron
lead
chromatogram
concentration
reactivity
hydroxide
indicator
soluble
safety
sulphur
amide
triple point
magnesium
endothermic
measurement
hydrogen
detector
iodoform
evaporation
chloroform
sodium
test
beaker
calcium
ammonia
insoluble
benzene
mercury
platinum
ethanol
alcohol
ammonium
bromine
pressure
volume
polymer
argon
mass
methyl
ppm

The course structure is very interesting and at the same time not very time consuming unlike other branches. If you attend all the classes, you won't need to study on your own. Majority of the faculty is highly qualified and gives a lot of input in making sure we understand. Nevertheless, you approaching the faculty will always have a positive impact.

You will realize that you are pretty much aware of what is being taught since it will be to a great extent an extension of what you have studied in +2. Courses comprise of the following:

Thermodynamics - Simple concepts. Subject involves First, second and third law which we all have studied in +2. A bit of extension which is obvious at a stage where you are studying Msc. Courses.

Organic Chemistry: Will mainly involve reactions and their mechanisms and in next semester the stereochemistry. Again an extension of +2.

Inorganic: Will include a lot of rote learning. But majority of it is what we have studied before.

Quantum: Totally a new subject. Will be very interesting and would provide with answers to a lot of stuff which went un-answered earlier. Watch videos online on youtube. You need to understand the concepts. For a normal person, if a particle is going from point A to point B in one single direction, it needs to cross the mid point. But thats not what actually happens in quantum. You need to come out of that classical thinking and understand the concepts. Videos will help you get clarity into a lot of concepts.

Labs: This is actually a very scoring component and can help you get your grade raised. Do NOT argue with instructors. This is the last course you would want to lose on the grade. Average gets B. Good learning opportunities are there in this course.

Second year will be the least hectic year throughout the span of your stay in Pilani. Take up formal/informal projects. You learn something really new. You get into the good books of professors, who can possibly provide you with recommendation letters when you plan to study abroad. Trust me the faculty is very well recognized. Also this is the year where you can actually score A's. A bit of sincerity and half the efforts of EEE/ENI students and you are good to go. Do not miss the tut tests as they will cause a huge difference between you and other batch mates.

Attending classes is the only thing you have to do and hey you get 10% marks for that in 1 course. Take Notes! You cannot read that 1000 page book any day! All questions will be from notes and that is something you should prepare properly by going for all the classes.


Economics
money
economy
future
people
teach
also
study
economics
life
save
spend
way
resources
important
world
works
learning
smart
use
good
make
teaches
get
years
buying
knowledge
land
crucial
People
invest
explains
need
system
just
going
trying
help
production
around
choices
understanding
keep
stocks
situations
predict
want
students
markets
ways
politics
making
affect
Econ
consumption
financially
revolves
selling
improve
like
market
learn
investments
shows
think
know
wisely
well
better
strong
try
live
take
show
labor
outcome
business
decisions
government
income
goods
includes
manage
things
great

While writing this, I have assumed that you haven't explored the field of economics/finance like many of us while registering for this discipline. I would also like to assure you all that your decision to pursue this social science programme is going to be one of the best decisions you have ever made.

So, like for other streams there is no skill set for you to gain at this point. My advice for you is to develop a love for what you are going to study during these upcoming years. But how?

1) READ A LOT: Start from the basics. The few of these are

a. Demand/Supply Concepts
b. Profit maximization
c. Role of government in the economy
d. Financial Markets
e. Financial Institutions and their role
f. Government's stand on various policies

2) ANALYZE WHAT YOU READ: While reading, various questions will cross your mind like

a. What causes inflation?
b. What factors govern the movement of exchange rates (like American Dollar vs Rupee)?
c. Why RBI increases/decreases amount of money in the economy?
d. How RBI prints money?
e. Why American Dollar is used for international trade and what are its consequences?

3) FOLLOW NEWS: You already know why this is necessary. Follow national as well as International news.

FINANCE SPECIFIC

4) FOLLOW EQUITY AND EXCHANGE MARKETS:

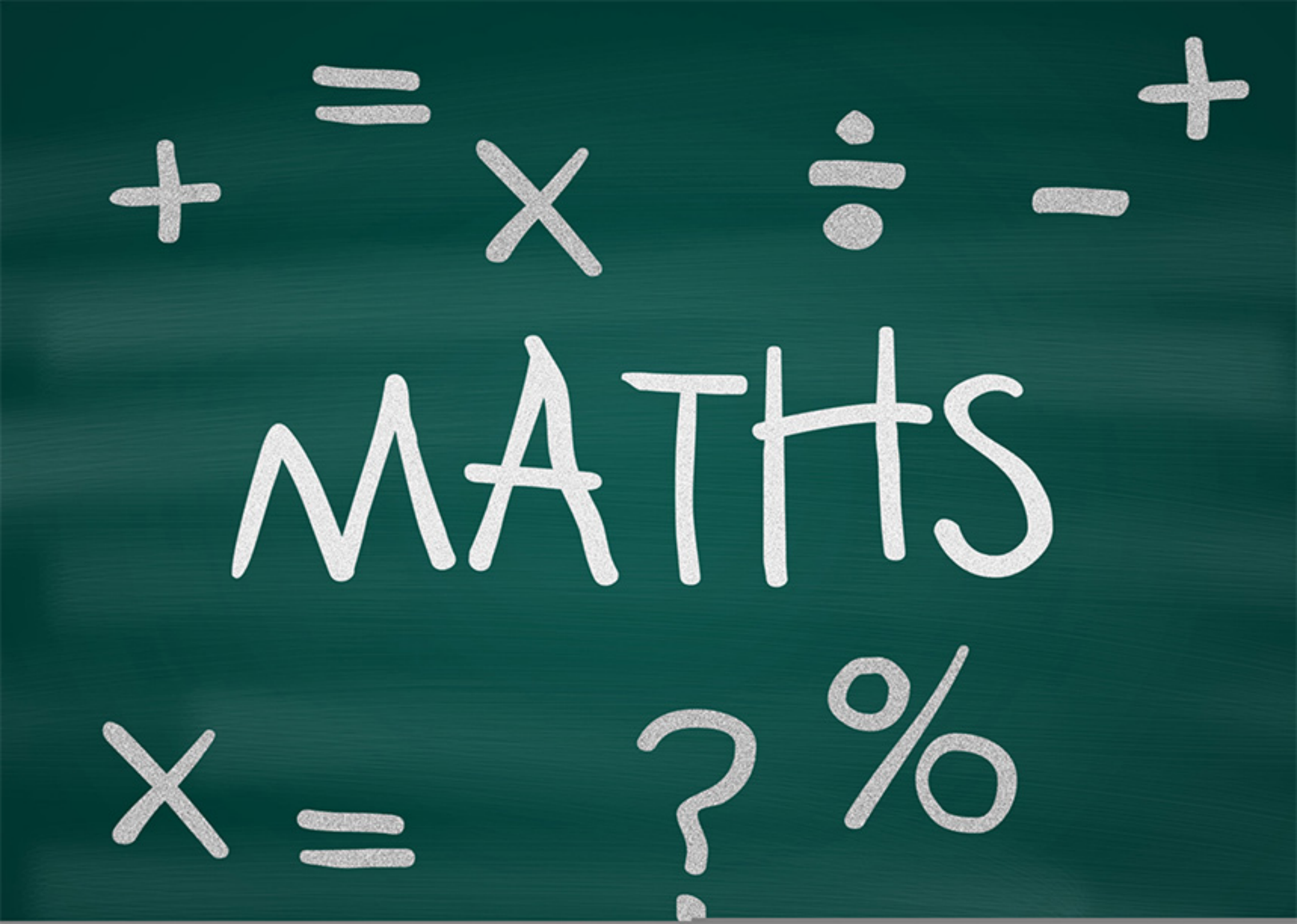
a. Learn about fundamental as well as technical analysis of stocks.

5) VIRTUAL TRADING: Create an account with nsepathshala.com or any other virtual trading platform and use your knowledge about stock analysis.

HELPFUL LINKS:

- http://www.investopedia.com/
- http://www.economist.com/
- https://www.boundless.com/economics/
- http://www.businessinsider.in/
- http://www.businessworld.in/

Good Luck !!


MATHS

1. You are actually into Mathematics.

1.You have prior experience/exposure (Olympiads)
First - Be thorough with the first year mathematics courses.

a) Multi-variable Calculus.

With your experience you must be able to say if you did the course properly or not.
If not, See the MIT lectures by Dennis Auroux. You shall be doing Elementary Real Analysis course in 2-1 , you don't want to be handicapped by shady understanding of the basics.

b) Probability and Statistics

Revisit this vital course if you are interested in the areas of Machine Learning, Data Science.
Get used to the idea of abstract definitions and problem solving,proofwriting etc.

c) Mathematics 2

Please go through the NPTEL course - Matrix Theory by IIsc Banglore. Must Watch.
It lays extremely strong basics.

$$F(n) = \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \end{cases}$$

So, after ensuring your basics are air-tight.
Start learning Higher Algebra , Real Analysis, Combinatorics etc.
Start working on problems that we have always wanted to try.

Utilise this summer to do our smallest form of research - try interesting tough problems.
There are couple of good resources -

a)International Tournament of Young Mathematicians. Worth our time and a lot more.

b) mathcamp.org

c) Explore the mathoverflow, mathlinks.ro forums..

Have fun.

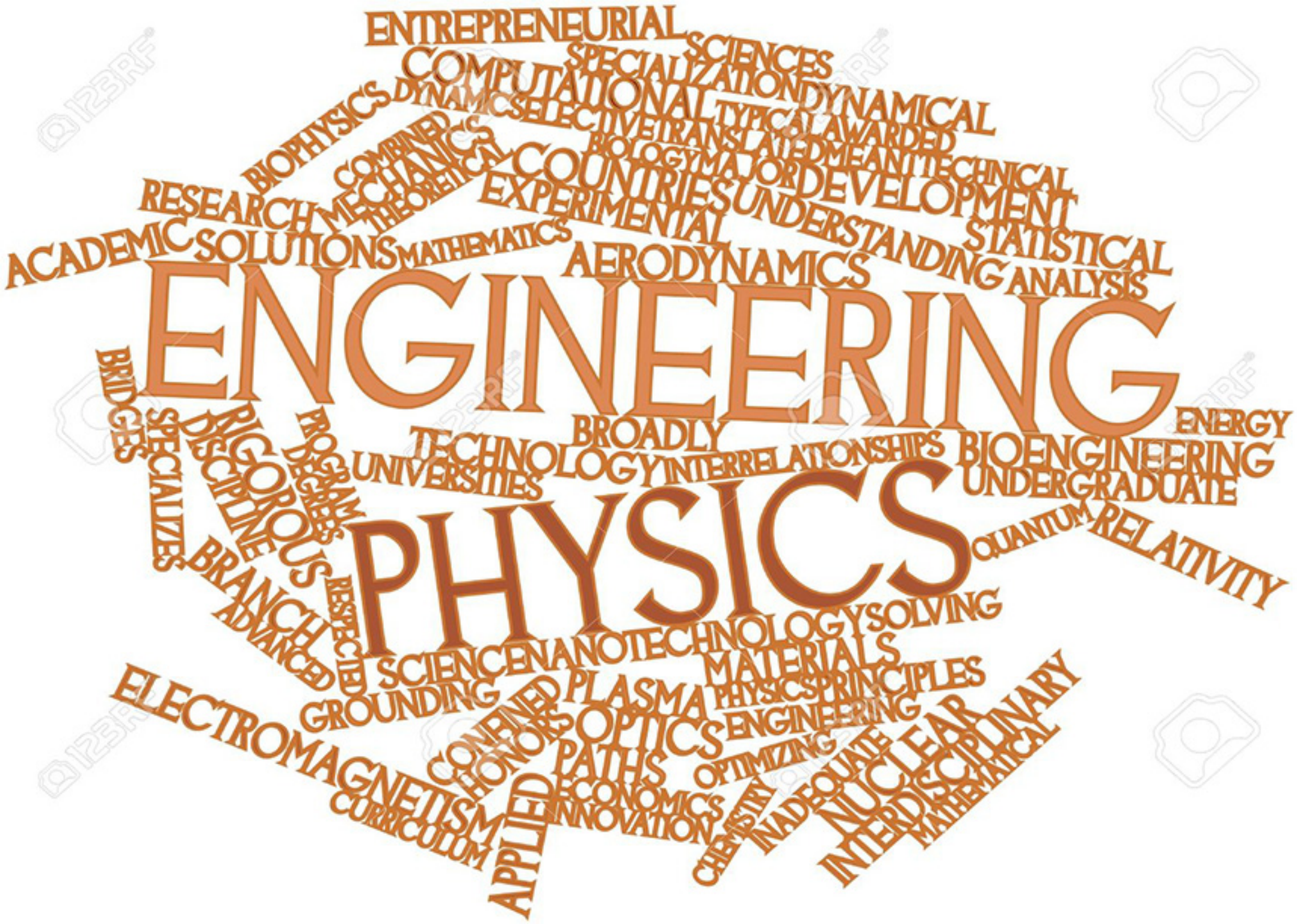
2. You are not into Mathematics.

Regardless of whether your CG is going to get you the stream of your choice.
Please do look at the desired Engineering section to start working.

You do not have prior Olympiad experience.
Become a member of mathlinks.ro and explore it.
Try interesting questions.
I suggest - Tournament of Towns.
Google it and get the question papers and try them.

Get accustomed to rigorous proofs. Look at creative proofs (Proofs from THE BOOK). Work/read through Terence Tao's Analysis 1 and 2. Look at different fields in mathematics - combinatorics, analysis, number theory, algebraic geometry, topology and so on. "Waste" time reading about them and exploring them. Get Math 1 and Math 2 basics right.

Priyans - If you want elegant problem (Mathematical) solving via coding - Refer to Project Euler.


ENTREPRENEURIAL
SCIENCES
SPECIALIZATION
COMPUTATIONAL
DYNAMICAL
TYPICAL
AWARDED
DYNAMICS
SELECTIVE
TRANSLATED
MEAN
TECHNICAL
BIOPHYSICS
BIOLOGY
MAJOR
COMBINED
MECHANICS
COUNTRIES
DEVELOPMENT
THEORETICAL
RESEARCH
EXPERIMENTAL
UNDERSTANDING
STATISTICAL
ACADEMIC
SOLUTIONS
MATHEMATICS
AERODYNAMICS
ANALYSIS
ENGINEERING
BRIDGES
ENERGY
RIGOROUS
DISCIPLINE
PROGRAMMES
BROADLY
TECHNOLOGY
INTERRELATIONSHIPS
BIOENGINEERING
UNIVERSITIES
UNDERGRADUATE
SPECIALIZES
PHYSICS
QUANTUM
RELATIVITY
BRANCH
ADVANCED
RESPECIED
SOLVING
SCIENCE
NANOTECHNOLOGY
MATERIALS
PLASMA
PHYSICS
PRINCIPLES
GROUNDING
ELECTROMAGNETISM
CONFINED
OPTICS
ENGINEERING
HONORS
PATHS
OPTIMIZING
NUCLEAR
INADEQUATE
INTERDISCIPLINARY
CURRICULUM
APPLIED
ECONOMICS
INNOVATION
CHEMISTRY
MATHEMATICAL

Physics is a branch that gives words to the beauty of nature and the way it operates. It represents the way we think and the way we interact. It requires an entirely different mindset that learns to appreciate things the way they are and then try to understand and use them for a greater good.

Try to develop an application based learning environment rather than solving on pen and paper. One must spend ample amount of time on each and every equation no matter how big or small it is and then try to apply it in real life either through a thought experiment or a demonstrative experiment.

Here's an example. Suppose one wants to understand formation of standing waves. Imagine yourself to be in a locked compartment of elastic material and you have a bag of elastic balls. Not you start throwing one ball to understand fundamental mode, two balls to understand first harmonic and so on. At a point you will realize that you have succeeded in understanding that nodes are energy exchange centers that help in interconverting potential and kinetic energy. Now this model itself has various implications some of which can be amazingly demonstrated.

The entire point I want to convey here is that a Physicist should be unbounded in his/her thinking and should have an independent thought process. This process may not always lead to correct models but that's how it works, wrong leads to right and right leads to wrong, but nevertheless the journey is exciting as always as it is full of visualizing nature in its purest form.

If you are interested in Theoretical Physics, you need a thorough base in linear algebra, multivariable calculus, differential equations and complex analysis. So yes, the whole of the foundational mathematics courses run in BITS. I'd suggest one to review all the math that was covered in the first year after the vacations. Trust me, mathematics is the language of Physics. You won't be able to do anything if you are not able to understand/use it freely.

Start learning Classical Mechanics (it is a course in 2-1 anyways, and the concepts developed here like general variables, Lagrangian formulation, Hamiltonian Formulation, Canonical Quantities, symmetries and conservation laws, Principle of Least Action, etc end up being used everywhere in theoretical Physics.

Spend a month or so trying to cover up all the material that I have mentioned. You can use Goldstein (the textbook, pdf/buy it) or try these notes from Caltech (you can skip stuff that you have already dealt with). Everything in theoretical Physics is done in terms of Lagrangians/Hamiltonians.

You need to know how to deal with particles in Classical Mechanics before you move on to how to deal with fields. There is no good book as such on dealing with fields. Before going ahead with fields, take a break and learn tensor analysis. The best part is that, you'll be able to learn the formulation of special relativity and general relativity easily once you know tensor analysis. Infact, you can do tensor analysis while you are doing relativity. Isn't that awesome?

After doing Classical Mechanics, try doing some relativity.

Covering Ch 1 in the book 'Spacetime and Geometry' by Sean Carroll will be enough for the summers. You'll learn to do special relativity as well as tensor analysis in the same. Also, at the end of the chapter, one can learn to deal with Classical Field Theory (which you should focus and emphasize on in your 2-1 by yourself).

It also shows you Maxwell's equations in a tensorial notation (this way of writing them down is only touched upon the Theory of Relativity elective) and you'll cover these (in their differential form, not tensor form) in EMT-1 in your 2-1 anyways.

So what do you learn? Basics of Classical Mechanics. Basics of Special Relativity. Brush with Classical Field Theory and Maxwell's equations. Also, enough knowledge to learn differential geometry (Ch 2 and 3 of S&G, Carroll) followed by General Relativity (Ch 4).

All this will surely give you an edge and make you comfortable. If nothing else, you'll find out if Theoretical Physics is where your heart lies or not.

Get used to big equations. Develop the ability to see "through" them. The best way to get started is Feyman's Lectures in Physics. The first two volumes are invaluable in developing intuition.

A good background knowledge of the relevant mathematics makes the understanding and appreciation of the physical sciences much easier and better, because that is the language in which physical laws are expressed with the necessary precision.

While calculus is the basic pre-requisite, certain parts of physics such as quantum mechanics, in particular, require a knowledge of linear algebra, including matrices, linear vector spaces, and so on. These topics are most useful in many other contexts as well.

A formal mathematical (but extremely well written) account is given in the book by Peter Halmos (I think the title is something like 'Finite dimensional vector spaces', but you can check this out on Google). The Schaum Series books on Linear Algebra, Matrices, Vector Analysis, etc. also provide good 'drill books' to learn the subject. Problem-solving is the best way to learn any of these things.

Entrepreneurship

- A great way to spend the summer is to do a few courses on Coursera. There is a wide range of courses from reputed institutes all over the world. One course I highly recommend is Introduction to Public Speaking (Washington University).

- Once you sign up for any of these courses, go to facebook and join their respective group(s). Over there, be a little active and make friends with the people there. This way you will create a very nice global network. Use the vacation to build a nice Facebook, Linkedin, Twitter and G+ profile.

- Starting a blog of your own is a very nice idea. Follow people's blogs and get into their blogging circles. Read Harvard Business Review articles on http://www.hbr.org/. Explore reddit. Explore medium.com.

- The best way to learn about entrepreneurship is to become a part of a startup! You need not have a very unique or new idea. Just find a problem and try to solve it. If you can't do so and still wish to learn, ask anyone from campus who has a startup to let you intern with them.

General

I can not stress more on the importance of having a blog. Once you start writing what you know, people will know that you know something. It is just an initial push which is needed, you need to write your first blog post and then you will automatically start documenting your work on your blogs. For reference see aishack.in (BITSian blog - 07 batch I think - Now in DreamWorks), jayrambhia.com (10 batch - got double digit no. of job offers because of this blog). You need to realize that blogging is a serious business. Technical blogs help everyone. Whenever you try to solve a problem, you look up someone's blog to find out a solution. It is your way of giving it back to the society. Secondly, blogging is a proof that you know stuff what you have done. No one can really question your work unless it is not authentic. As I said earlier, it gives you a tonne of job opportunities too! Blogs need not just be about how you code, it can be about cool mech or electronics projects, or some theory you think is interesting which you would want to discuss with general public, or it could be about describing your take on a particular theory.

If you wish to write creative stuff, like poetry or short stories, or even take on current political or social issues, you should ideally do it on a separate blog. Getting a blog is totally easy and free. Sign-up for a wordpress domain or a blogspot domain. If you have cool demos to show, start a Youtube channel.

Have some confidence in yourself. Confidence is lacking in many people and that is the cause of why people don't end up doing anything. So, Have some faith and confidence in yourself. It is not necessary that you must end up liking whatever you start off with, it is more important that you do stuff and end up gaining valuable experience from the same. Every activity teaches you something. So, don't be another smart person who is busy convincing yourself that you possibly don't know enough to do anything/isn't smart/of no use. Go, do something. Fall, get dirty. You'll eventually fall in love and forget all the bad days.