**Experiences and then integrate - starting from yesterday’s tm and movie:**

**Consider adding self portrait below at end of second page:**

**Put in management infrastructure**

**Add in TM notes:**

**Top-level picture view:**

**Goal: Working version / production:**

**Integrate experiences into sheet**

**TODO: add labels**

**TODO: modifications**

**Todo: add ring**

*(TODO for Management: 1. check ordering, including where leadership goes- 2. Put below personal development? 3. Figure out where leadership goes 4. Connect to either Pers. Dev. Or Management endpoint to comm.)*

TODO: measure of how good

TODO: know to balance the following: happiness/confidence, intensity, quality

**Summary:**

*Formation and Themes:*

Be natural, and then work hard too.

PC/P: learn yoga to its finest.

S and m: / Think about Hilton Garden Inn, where you felt it first. MIT.

Reset state: clear mind

Start state: ohm world is s and m

Pin down center:

Don’t lose yourself, but also improve yourself.

How this was formed: Keep it simple, research questions, integrate, comm, logic.

History+ levels: Maturity: Amanda: | Money: | Stress relief: | Gropp: | Mgmt: habits-summary | PNA: |PCA: | OOP: |Dad: |Man from internet |Experience: mom|

Cross-link:

Habits summary is a preview of maturity, just before 3d cubes. Helps to smooth cubes out. Should always be thinking about the worklife from starting man from internet and dad, and with management on top. Static can be smoothed out through maturity lines.

MPI book: stick.

MPI Implementation: bschool.

MPI Standard: engineering, social intelligence.

*Rules:*

Habits need to pervade everywhere throughout.

Vertical in wl. Horizontal in wl.

Rule propagation: example: confidence – think about how someone else responds.

Use logic to make things follow. Example: keys, wallet, cell -> backpack, etc.

Experience Generalization to rule. Example: clothes in India, generalize to many things.

Experience transfers to other experiences.

Recurse: this generally happens naturally, but just know that it can be done.

*Prioritization:*

Work: Code: 40, Data:30, Admin: 10, Writing/pres: 20

Mgmt: Goals: 30, month: 10, weekPlan: 30 , daily todo: 15, 10: prioritization, 10: scheduling

Comm: L0: 40, L1:25, L2: 15, L3: 10, L4+L5: 10 Wl cheat sheet.

*Implementation:*

Practice: remember calculus exam study. Once you practice and memorize, it becomes in-grained in your head in working memory.

experience-sheet of paper with situations, mgmt: ppt and latex. use 4 sheets as a guide esp. when static comes in. Look at implementation picture. Habits(expand on this – needs to pervade all). When applying or doing an activity, use number system to identify what part it belongs to. Independence/Interdependence.

Discuss how every week should be handled in terms of week plan doc and goals doc, along with sheets.

Independence:

1. proactive: circle of influence vs circle of concern --
2. begin with the end in mind : goals.
3. prioritize: urgent/important, deadlines.

Interdependence:

1. Think win-win: --
2. Understand, then be understood: --
3. Synergize: --

Quality:

7. Sharpen the Saw: practice, re-evaluate system, read a book and come with it at different angle.

Concrete implementation: (in terms of files on comp)

Sharpen the saw.

*General wisdom from going through the sheets:*

Money and work are close, but work and happiness are even closer. God and happiness and Social intelligence and maturity are connected.

Money stays constant, and is related to static. Comes through experiences. Sayings.

**Follow calendar of events (full implementation):**

General Todo:

1. Prioritize the above.
2. Break each into smaller pieces.
3. (Maybe) Look at competition, and do better.
4. Make sure to maintain minimally working version of you.
5. Have full pathway of entire sheet intact (can be multiple different lines).

3. Do these in sittings when there is nothing else to disturb.

6. One thing at a time, and add.

7. Order the additions of different things to add.

*(note: add more, org better, make plan to read newspaper – gen Knowledge + 52 weekends nytimes books).*

TODO: (1. Shorten this 2. add more to comm 3. Make Work section better 4.)

Quick Todo: 1. Fix the experiences section, going through experiences. Add real reliability points. 2. Add real happiness – 3 hours. 2. Figure out 3-D; 3. Prioritize worklife cheatsheet. 4. Add text to guide each section. 8. Add tm sheet, w/ social int sheet – Tuesday. 5. Add text to guide each section in mgmt. 6. Add worry management to happiness in habits. 7. Consider margin changes for more space in wl-cheatsheet9. ../thePast/maturityVersions: Clean-up text, make accurate, add intermediate levels; 10. Read habits – confidence integration; 11. Order content, make any points clear, order/structure content, connect all points together. 12. Add in happiness wiring, and connect all the way down to real-time confidence 🡪 static to dynamic. 10. Make sections in above more clean. 11. Fix the section for gen knowledge. 12. Integrate summary with cheat sheet better.

*God:*

Note past situations from childhood that could create awkward situations and define relationships. – could put this in experiences.

Remember that all acts as guide, especially at the top and bottom (gut /randomness is ok). Randomness can be used to your advantage.

**Personal maintenance:**

* 1. *Regular Routines (habits)*: (add notes from habits book) check nails, comb hair, belt on / pants high, brush teeth, shower, shave, lotion, concealer, charge phone/laptop, fill gas, clean out backpack, workout, check for events and calendar, check email, get clothes for week.
  2. *Static:* link to 4 detail sheets:
  3. Habits summary *🡪 Ecological Intelligence -* impact region, economical.
  4. *Happiness (apply to happy thought): urbana house. If you think about girl, think that there’s no doubt...*
  5. *UnderstandWorry:* emotional intelligence: whatAreEmotionsFor+ScienceOfEmotions+Emotions / Intellect disparate, self-awareness/real-time awareness, anger+anxiety, delayed gratification + positive thinking + Hope + Flow-confidence / learning and Flow
  6. *StopWorry (apply to each worry):* day-tight, what’s the worst, knowthefacts, prob/cause/all sols/best-sol, don’t saw sawdust, cooperate with inevitable, don’t imitate others, don’t get even, meditate/exercise, (get books on self-confidence). If someone asks question, don’t think negatively.
  7. *Work:*
     1. Sincerity, ethics and honesty, Diligence, work 80 hours a week, enjoy what you do.
     2. Prior work, baseline data, define problems, contributions, theoretical analysis.
     3. (P)NA. algs, Code/Impl, Apps, Research Questions + Results, Profiling to Explain Why (App, hardware/PAPI, metrics). (P)CA.
     4. Project management & collaboration (git for code, cmake, README).
     5. Writing skills (latex for presentation, svn for paper collab, elements of style, persuade through intro, no formatting mistakes, envisioning information, presentations, posters).
     6. Awareness of technology to be using in your field / Awareness of developments in your field – related work.
     7. Time management at work (meetings, schedules, deadlines), Administrative.
     8. Marketing yourself and your work – title, abstract (dry/factual), intro, put info on website/linkedin;
     9. Work:rel:admin:
     10. Companies: BNL (DE. Shaw, GS), PNNL (Yahoo, Google, Palantir, fb, cray), Stanford, LLNL. Keep in mind that all companies are good (it’s what you make of it).
     11. Context/Weather/environment:
  8. *Personal Development:* 
     1. Depth of Gen. Knowledge / hobbies: Tennis, Piano, Bridge, chess.
     2. Leisure: automotive, tv shows (Office, MF, Friends), songs.
     3. Financial: economics, stocks, gas prices.
     4. Current Events: Books (World is Flat, Hot Flat and Crowded), magazine(), News(work: wired, gen: nytimes),
     5. Logistics: city structure, airport structure, general driving knowledge, grocery store structure, gps, subway systems.
     6. Geographical: U.S. map, europe map, city maps, airports, freeways, quirks about places, subways, landmarks, driving.
     7. Work/professions: lawyer, engineer, scientist, educator, doctor, hr.
     8. Geo-cultural/political: cultural, political (liberal vs. conservative, practical vs. theoretical, gen. vs. spec.), climate/weather.
  9. *Management:* 
     1. Health: Weight (chew food, avoid eating large portions when available, Exercise: lift weights, running, yoga), Avoid sickness, Acne, Dental (floss, whitening).
     2. Money: southwest fares, reimbursements, rent, uiuc bill, cash, paycheck, account for cash, owing ppl money, points for air travel.
     3. Time: Look at calendar for each week, write action items in iPhone todo, have monthly plans, find social events/tech events, prioritize.
     4. Spaces: Jean pockets (KeyWalletCell, m-*ccnum*), Backpack (don’t put on floor or bed), office, laptop (clean outside, org folders), room (bed, closet, floor, lights, clean), car (outside, carpet, inside glass, m-lic. Plate, usage-cost/others), kitchen, bath (floor, towel bar), lock doors before leaving, airfare.
     5. Appearance: belt on, pants high, XYZ, deodorant, gum, laundry stains, shirt not inside out, iron shirts/pants, button shirt, matching/non-torn socks, plan weekly wardrobe, check for clean teeth and face, non-inverted shirt collar, shirt not half-tucked, haircut/nailcut, hairgel, dinner etiquette.
     6. Facebook/LinkedIn/twitter/instagram/WhatsApp: know software updates, profile up-to-date, check frequently.
     7. Leadership: Listening, critical thinking, giving feedback, time management, planning and implementation, organization and delegation, facilitation, motivation, mentoring, team building.

**Social interaction:**

* 1. *General principles:*
     1. Communication organization:
        1. email: check recipients, no double-sends, check spacing/formatting, spell check, check for internet connection before sending, don’t send angry emails.
        2. gchat: type outside chat box, send quickly and have a roadmap for conversation, be clear, check spelling.
        3. text: short texts, quick replies, no getting overworked about someone’s confusion about you.
        4. phone: articulation, speak loudly, leave short vmails.
        5. skype/video: find a place with good lighting, get audio/video tested, make sure background noise is out.
     2. General rules of interaction:
        1. Level 0+1: Think positively about interaction: No getting overworked abt other’s remarks, stop beating yourself up, be humorous / Be Honest, ethical to yourself and others | Act cozily | Reliability / Consistency / Confidence /Shout.
        2. Level 2: Be logical, make sure you make sense / avoid random thoughts that come to mind / structure thoughts and check for non-sequitors.
        3. Level 3: Be intentional / Think before you speak / integrate with top-level point.
        4. Level 4: send message in way that others will easily receive it: *howtosayit*: grammar/picture errors, emphasize at the end, that vs. which, check for gen. rules (don’t use abbreviations), *bodylanguage*, *vocalvariety*.
        5. Level 5: *persuasion, avoid I: a. give honest and sincere appreciation b. don’t criticize, condemn, complain c. arouse in the person and eager want:* Be happy with others: Avoid mis-interpreting others’ intentions / don’t get mad when someone doesn’t understand you | avoid neg. thoughts | Identify common things | Listen to and Read people (body language, facial expression) || *empathy, wired to connect:* Situational awareness / Focus on one thing, and do that thing / Make eye contact / don’t be interested in random parties or randomness outside / emotional intelligence | Give space, avoid being over-intrusive.
  2. *Specific settings:* 
     1. Technical presentations: find ways to break ice and avoid nervousness/think of audience as rooting for you, organize speech (be linear), get to the point, how to say it, your body speaks, vocal variety, research your topic (know content), use visual aids/materials (concrete?), persuade, inspire/connect, entertain/joke.
     2. Social gatherings/conventions (table topics): Be resilient to others words, think of words before speaking and making your point, punctuality, prep clothing, know schedules and email others in advance, know interactions that may come up, know what to go to/where to be, identify common ground in group and situational awareness, connect with the overall situation (mtg theme).
     3. Meetings/hangout/interview (conversation): prep for what to say, avoid going into things that generate unneeded work.
     4. Relationship setting: Be on lookout / think positively / icebreaker, Planning dates (make reservations, tell ahead of time what’s going on, don’t make things too complicated), Have something interesting and unique to say, show interest in specific things, Connecting (Find common ground, show compassion, show interest in the other person’s activities), Express emotions when you have them (when you like someone, tell them), connect about emotions.
  3. *Social Intelligence:*  --- 🡨 habits summary.
  4. Mannerisms to have and to avoid: project voice, talk slowly, lips relaxed, smile, no frowning, eyebrows up, relaxed hands, avoid over-smiling, avoid moving arms, hands in pocket or to the side, avoid crossing arms, back straight, walk with brisk pace, don’t lean on places, walk straight line without feet scrubbing floor, give hugs tightly, shake hands thoroughly, no looking at phone, avoid bodily noises (no teeth clicks, no nose sneezes, no farts), don’t bump into people, clean trash behind you.
  5. *Working with Emot. Int (fix):*experience: emotional, social, general if/then, reliability, maturity/practicepoints/positivepoints, rule-experience, specific if/then, security, fault-correction, fault-tolerance.

**Application of Pers. Maint. And Soc. Interactions to Real Situations:**

Method of learning, and validating model

1. Identify situation, analyze what you did right and wrong
2. Write it down in docNotes
3. Practice it again in situations
4. Figure out where it fits in workLife -- connect with big picture
5. Unify, simplify and generalize
6. Some things from situation carry over to other situations
7. Organize according to general rules of interaction
8. Identify upcoming events and apply based on general rules + specifics
9. Prioritize the items

***Any situation:***

* belt, pants high,
* check wallets/key/cell , shave
* iron shirts
* deodorant
* no stains
* go to bathroom
* check concealer
* comb hair
* back straight
* think of points / objectives
* driving/transport

***Rel: Weekend Date:***

* go running
* Emo. Int.: identify points of nervousness,
* Self-awareness:
* Self-regulation:
* Motivation:
* Empathy:
* Social Skills: influence, communication, conflict management, leadership, change catalyst, building bonds, collab, and cooperation, and team capabilities.
* If she’s poking fun at you for your weirdness, don’t get stressed about something someone said, or something you said, don’t get nervous, don’t get worked up on an opinion.
* Comm: have topics to talk about to open conv. (plan this out), eye contact, avoid distracting mannerisms.
* Comm: inspire: find common ground on little opinions, don’t talk about weather or generic things like work/traffic/news, find a place to spend time together alone.
* Mgmt: Time: Plan events for yourself a week in advance.
* Mgmt: SharedTime: Tell plans to other person, make sure it works for them.
* Mgmt: Time: Talk to parents about it / talk to friends about it 3 weeks before hand.
* Mgmt:appearanceVuln: no buttcrack, don’t smell, teeth/lips clean, socks clean/non-torn.
* Comm: lips relaxed, smile/avoid looking stressed, back straight.
* Mgmt: Appearance: Figure out clothes to wear each day, concealer
* Mgmt: Spaces: Make sure phone fully charged for each event
* Comm org: make sure to send to correct recipients, be clear in texts during coord.
* Mgmt: Logistics: know directions, know how to get to/from places.
* Mgmt: appearance: Dinner etiquette at meals.
* Error-detection: watch for
* Error-prevention:
* Error-correction:
* fault-tolerance:

***Rel: Skype Date:***

* Comm org: Make sure media is fine, find good lighting, have an intro.
* Comm:org.+intentional: have goals and outcomes, Figure out topics to talk about which connect.
* Comm: Make clean exit, know how to finish convo, keep convo to 30 mins.
* Comm: Mgmt: Ensure no one is around.
* Comm: Mgmt: Get the timing clear (time zones, etc.)
* Weddings, dinner parties, bar meetups, house parties
* Don’t laugh too much at something, to give the wrong impression

***Meeting with a friend at a bar:***

* Comm: Have specific items to talk about and catch up on
* Comm: Know answers to direct questions.
* Comm: Warn yourself to not be too intrusive.

***Meeting person X at work:***

* make sure you don’t smell.
* avoid farts, go to bathroom beforehand.

***Meeting advisor***

* have clear points to communicate
* make preview before hand
* be linear

***Meeting with several friends:***

* Comm: don’t be too intrusive.
* Have opening:
* Comm: don’t be excessive.
* Check shirt and make sure things are in order with quick changes
* Avoid negative points.
* Don’t say something and modify.
* Know what to do when you meet certain people.
* Don’t be too intense, esp. if someone is talking about something familiar.
* Don’t be the center of attention.
* Quick points without
* Say a few things, don’t go silent.
* Don’t get worked up when someone says something
* Don’t try to get attention in a group
* Eat with proper etiquette
* Pay attention to what’s going in a group setting, if questions, then ask
* Have closing

***Group Meeting:***

* Mgmt: send email beforehand to discuss what to do
* Make sure to discuss results, action items
* Comm: don’t discuss too much

***Full-body /all-hands Group Meeting:***

***Party at a bar:***

* Know directions, figure out timing for how long it will take
* Find people to go with
* Get the right clothes to wear, make sure clothes are ironed
* get cash

***Party at friends place:***

* Mgmt: find directions to house, check that you can get in, if shared community
* Comm: Be quiet and don’t impose
* Do what others are doing.
* Offer to help
* Don’t overstay your welcome

***Work social hours:***

* take badge with you
* avoid over-eating
* talk to people about your work
* clean shirt
* go with clean jeans
* avoid looking like you need the food

***Wedding of a close friend:***

* Mgmt: know clothing
* Have a list of sub-situations
* Know the agenda without intruding on others
* Don’t try to get attention from others

***BMM Convention:***

* Check events beforehand
* Don’t try to get attention from others
* Clothing

***SC conference:***

* check events beforehand
* clothing

***Lab Presentation:***

* don’t make assumptions on what people know
* be ready to answer questions easily

***Coding:***

* Follow code complete coding principles
* Think about big picture
* Compilation procedures
* Google when errors occur, and when confused
* Take breaks

***Results Collection/Experimentation:***

* envisioning information
* talk about which research questions you are answering
* think about research question

***Paper writing:***

* Correctness in techniques, problem and results, Make sure about words making sense
* Structure is logical and flows without forward references
* CommInWork+Mgmt: commit changes and let people know
* Overall structure placement is good, submission-ready.
* Consistency across sections
* Use the right vocabulary
* CommInWork: Writing style

Specific Scenarios:

***Coming back home from the car:***

* Put keys, wallet, cell phone in the right place

***Email exchange for work:***- Check email formatting

* Use good style – read how to write good emails

***Someone asks “how old are you?” or personal question. In a group setting where friends getting attention and you’re not.***

***If someone gets angry at you:***

***Meeting a girl who likes you:***

***What do you work on?***

***Anytime:***

***Applying experience to above:***

***Implement the experience-learning:***

***Fluff( ) ; implement experience-learning.***

***Fluff2***

***Reliability : /error correction spinning with experience – spinning with practice.***

***What one person might say can apply to another person, though to a different extreme.***

***Make sure situations are clearer (why didn’t I bring the lexus?)***

***Cheating: teacher thinks cheating a regular failing student, but student doesn’t believe.***

Work: Working with Tami/secy,

Social: Honda dealer car.

Work: Gropp, bronis, …, garzaran, padua, torsten, todd,

Family: Dad, mom, Atul, Ridhima, Rakesh Uncle, Pinki Auntie,

Friends: Rishi, Chris -

Collaborators: Amanda, .. Costin, …,

Related Work: DPLASMA, rice-sarkar, locality,

Competition: … , …,

Work status: (layers): Livermore scholars, students

Social status: Indian, Brahmin, Marathi, …

Bosses: advisor, students,

Relationships: sateja, avani, others in queue – remember asm, experience.

Fine-grained.

Real-experiences – Paris.

God.

HIVE, count your blessings, harness the positive thoughts, alleviate key points of stress by talking them through, YOLO, dwell on happy thoughts (get books on depression). Urbana house, sunysb, -, highschool grad; asm house.

**Happiness:**

(read intro)

**Stress Reduction / Anxiety / Meditation:**

1. Live in day-tight compartments

2. If Trouble backs you in a corner:

a. ask yourself, what is the worst that could happen?

b. Prepare yourself to mentally accept the worst if necessary

c. Calmly try to improve upon the worst, which you have mentally tried to accept.

3. “Those who do not know how to fight worry die young”

4. Get the facts. “half the worry in the world is caused by people trying to make decisions before they have sufficient knowledge on which to base a decision”

5. After carefully weighing all the facts, come to a decision

6. Once your decision is carefully reached, act! Get busy carrying out your decision, and avoid anxiety of the outcome

7. Business worries:

1. what is the problem?
2. what is the cause of the problem?
3. what are all possible solutions?
4. what is the best solution?

8. Crowd worry out of your mind by keeping busy.

9. Don’t let little things – small bugs/beetles - ruin your happiness.

10. Use the law of averages to outlaw your worries. “What are the odds against this thing happening at all?”

11. Cooperate with the inevitable. If you know a circumstance is byond your power to change or revise: “It is so; it cannot be otherwise”

12. “stop-loss” order on your worries .

13. don’t saw sawdust. let past bury its dead.

14. Fill our minds with thoughts of peace, courage, health and hope. We control each thought. Our life is our what our thoughts are.

16. Never get even with our enemies.

17. Don’t expect others to be thankful to you.

18. Count your belssings not your troubles.

19. Don’t imitate others . “envy is ignorance”. “imitation is suicide”

20. Life hands us a lemon, make lemonade

21. Forget our own unhappiness by trying to create happiness for others.

22. Be spiritual

23. Unjust criticism is often a disguised compliment. Remember that no one ever kicks a dead dog.

24. Do the very best you can, and then put up your old umbrella and keep the rain of criticism from running down back of your neck

25. Keep a record of fool things we have done and criticize ourselves. Since we can’t hope to be perfect, let’s do what E.H. little did: let’s ask for unbiased, helpful, constructive criticism.

26. Avoid thinking of others comments as trying to hurt you

27. Try not to think too much about one thing

**Getting things done (?)**

Know when something is adequate and get it done.

**Work:**

1. Sincerity, ethics and honesty, Diligence, work 80 hours a week, enjoy what you do
2. Systems issues for setup: OpenMP runtime, moab scripts, compiler setup, .bashrc, ssh guide, machine access/banks, emacs editing, ssh key, macports, ‘use’ rose, use adept-utils
3. Baseline data: slack data histogram, noise histogram, increasing slack as we scale, dequeue overheads, other sources of noise (e.g. software error correction).
4. Develop theoretical analysis: Perf. Model based on baseline, engineering sched. parameters, execution time, validation of model with basic results, cost model.
5. Algs: apps/teasers, numerics, NP-comp, recursion, sort/search, DFS/BFS, BST, stacks/queues, lists, arrays, bit ops
6. Basic code/Implementation for stencil/dotproduct (make sure all optimizations finished, need to apply theoretical analysis here):
   1. Fortran:
   2. C:

v. App programmer usage + techniques: ROSE source-to-source, documentation of what the programmer has to do, use cases/corner cases when it doesn’t work, counting the number of MPI+OpenMP regions.

1. Slack-conscious sched runtime: vectors, libunwind, wrap.py, slack pred strategies, overhead, error, noise/dequeue measurements, (optional: use additional slackpred methods).
2. Cmake software management: linking shared libs, high-precision timers, libraries (ssl, unwind, math), software arch.
3. Git: git stash, know commands, understand what’s going on, understand branches
4. Research Questions to setup experimentation:
5. Data points and .dat files org:
6. Data processing scripts/code: use python arrays, can use code from C here as well
7. Plotting scripts: .py scripts, organize plots
8. Processing results text output file of app run:
9. Apply techniques to actual Apps:
   1. NAS LU: ensure makefiles complete, update code
   2. AMG: run the amg files
   3. PF3D: run .i files
   4. Pcm:
10. Performance results validation: Graph on varying sf, varying slack, varying noise, varying task sizes .
11. Profiling to Explain Why it works: Costs of idle time and dequeue overhead
12. Persuade through intro: explain things one at a time, guide readers, state assumptions, “Elements of Style”.
13. Literature search / related work: DPLASMA, Stanford work
14. Weekly Email updates: use good email etiquette, “elements of style”
15. Time management at work (schedules for projects, meetings, deadline): send out plan for year
16. Awareness of technology to be using in your field.
17. Marketing yourself and your work
18. Awareness of other people’s work/competition

Situations:

1. Ensure that the correct result is put in intro and abstract.
2. Latex:

- svn commit the .tex file with updated pdf generated from python script, and make sure to say something meaningful for it.

2. Python script:

a. Layout:

b. graph style and presentation: look up the best way to present the data (envisioning information)

speedup:

svn commit the script, svn commit the pdf generated and update the file in the latex as needed, making sure it looks correct.

make sure bash environment is setup correctly to have python from macports and pdflatex from macports

make sure you have the right .sty files and the correct paper format files

.dat file : put numbers in column

commit the .dat file

read numbers from the runtime, and print to .dat file: python script

scripts for running code: set environment variables, make sure it works for the particular machine in question

runFortranCodeVarySF, runFortranCodeVarySF.csh

Specific Graphs:

1. Scalability Line graphs

- Make sure that lines don’t get squished, adjust scale

- check that all lines show

- Trends for each strategy on NUMA and non-NUMA .

1. Speedup graphs

Dos and don’ts : put everything in central spot

Spreadsheet for jobs

Job search apps, Paper? , Project management software

Make a schedule: work for 30 minutes a day:

Make templates:

Mgmt: rel: vulnerability: Back it up: (use box)

Todo: next steps

Job postings, deadlines, application methods, hiring priorities, application documents, organization research, reference information, follow-ups, application status , requests for add’l materials , timelines, networking contacts, interview prep, post-interview notes, contact info, email initiation

To do: next steps: what are 3 things you can do in the next week to get on track to stay organized?

1. Work: admin: Create a word document with one full application cycle.
2. Work: admin: Create an excel spreadsheet to store the different apps
3. Work: admin: Read interview book to get additional information to shape. And talk to go.illinois.edu/GradCareersAppt/

***General Goal for Work***: Graduate in 6 months.

Very long-term goals: My PhD thesis work had focused on low-overhead multi-core scheduling techniques for use in the context of scientific and high-performance computing. My long-term goal within, say, the next 10 years is to use the research I have done during my PhD to make an impact to the field of computer science through offering new perspectives to a particular area of computer science or to develop an exemplar technological solution to a problem. By being selected to attend the forum, I could have a unique opportunity to help me achieve my long-term goal in three key ways.

First, I could learn from experts their opinion about future trends in technology, which will help me to better anticipate and navigate the future trends that I would likely encounter through my career. For example, given current advances in quantum computing research, what is the potential impact of novel quantum computing technology to my PhD research topic, within the next 5 years? Could I make modifications to the techniques in my PhD thesis so that they could provide benefits in the context of a quantum computing technology? Knowing the answers to these types of questions would be beneficial to strengthening the thesis work I have done, and to apply my work in the future.

Second, I could learn how to formulate the problem in my PhD thesis mathematically. This forum is unique to me compared to conferences I have gone to up to this point, in that it brings together mathematicians and computer scientists together in one forum. By discussing and learning from experts in mathematics how to formulate my thesis research mathematically, I believe that I can more effectively apply my research to areas outside my area of specialization. An additional benefit of talking with experts in the field of mathematics is that I will be able to increase my network to mathematicians, rather than restricting my network to only computer scientists; this could allow me to potentially form new types of research collaborations, which are mathematically-oriented, in the future.

Third, and perhaps most importantly, I could understand problems in society that are impacted by my dissertation research by learning from experts how problems in society can be impacted by computer science and mathematics in general. In the context of my research, could the low-overhead multi-core scheduling techniques in my thesis be used for developing energy-efficient operating systems for laptops used in developing regions of the world? The forum’s discussions would enlighten me on how to implement this in the near-term, and learning from experts how to apply my thesis work to real research problems in society would help me apply my thesis to real-world, practical problems in industry that I may encounter throughout my profession.

Overall, learning about experts’ viewpoints will allow me to formulate a well-informed and well-rounded vision that I can use for advancing myself in the next several years of my career. The forum is also relevant to my current research directions, and near-term goals. For the above reasons, I believe I could be a good candidate for the Heidelberg Laureate Forum, and hope to be accepted to be part of the Heidelberg Laureate Forum this year.

Resume

Cover Letter: Please tell why you would be a good fit for the position

Reference letter by Prof Cappello:

Reference Letter by Prof Gropp:

Reference Letter by Prof Garzaran:

I am writing this letter in support of Vivek Kale for his application for invitation to the Heidelberg Laureate Forum. I have known Vivek Kale since I moved to University of Illinois at Urbana Champaign as a visiting professor in March 2009 to establish the INRIA-Illinois joint-laboratory on peta-scale computing at NCSA, and have had a continued relationship with Vivek throughout his PhD. Before moving to Illinois, I led a team of 30+ people with up to 7 permanent researchers, working on High Performance Parallel and Distributed Computing. In particular, I published at the ACM/IEEE Supercomputing conference in 2000 one of the first papers about hybrid MPI+OpenMP programming of cluster of multi-core processors, a novel method of implementing a large number of numerical simulations and scientific applications. I remain engaged in this area.

The MPI+OpenMP programming model is currently one of the most popular ones to use for scientific simulations intended to run at large-scale on modern supercomputers. However, although MPI+OpenMP implementations have made outstanding progress since 2000 on a large number of current supercomputers of 1,000 processors (or, nodes), managing the OpenMP threads and their workload efficiently still remains complex as scientists scale applications to 10,000 nodes or more. The problem comes from the fact that bulk synchronous execution is used by many HPC applications, due to the domain-specific constraints in the application, which cannot relax the synchrony of communication. Given this, if there are small performance fluctuations due to operating system events (which we refer to as noise, or interference), in an MPI processes within a particular time step of a simulation, that noise can amplify, producing a major performance slowdown at large scale. This problem was first established in 2003 by Petrini et al. Much work has been done to quantify and understand the impact of system noise (Tsasfir ICS05, Hoefler SC10), showing the continued and increasing importance of this problem. A second related problem is that within-node application-level imbalances are a significant factor in total imbalances across all cores of the machine, and the contribution can be larger as more cores per node exist on next-generation supercomputers.

Some low-level strategies such as co-scheduling, as suggested by Petrini, have been used to mitigate impact of system interference, but have been limited in their performance impact, and also do not handle within-node performance variations induced by non-OS events such as transient hardware variations. Additionally, within-node application-level imbalance can be handled by measurement-based load balancers such as Zoltan or Charm++, but this is inefficient and challenging to do because its mechanisms are inefficient. To address both of these problems together, Vivek followed a new strategy: develop an intelligent mixture of OpenMP static+dynamic scheduling so that the scheduler overheads could be reduced and adaptivity could be increased, both of which can help to significantly increase performance. The proposed approach consists of experimental tuning of various mixtures of static and dynamic scheduling, so that within-node imbalances can be mitigated without allowing the overhead of dynamic scheduling itself to extend the critical path of application execution. Using these scheduling strategies to balance will significantly help to reduce total application imbalance across all cores. The resulting paper that he wrote on this scheduling strategy was his first published paper and received a best paper award at the EuroMPI 2010 conference held in Stuttgart.

Vivek then continued this work towards a Ph.D. under the supervision of Bill Gropp. He first focused on his mixed static/dynamic scheduling strategies for dense matrix factorizations. Dense matrix factorizations are a class of numerical linear algebra computations that are highly time consuming in many engineering applications such as the design of an airplane wing. Vivek worked with Simplice Donfack of INRIA to improve the performance of a communication-avoiding LU code and obtained a 35% performance improvement over a standard DPLASMA implementation, which is the standard implementation for LU factorization at the time. Simplice and Vivek worked together to improve performance of these schedulers for a variety of matrix sizes through a careful tuning of the mixture of static and dynamic scheduling. Notably, his schedulers achieved higher performance than two widely /used industrial implementations of the same algorithm. These performance gains were done with the help of a performance model and theoretical analysis developed by Vivek. This is an important result since communication-avoiding LU code is already highly optimized on multi-core architectures. Vivek then published a paper with Simplice Donfack to discuss these results in IPDPS 2012. He visited INRIA France to work with Simplice (under supervision of my colleague Laura Grigori) on continued work for one month. This collaboration shows Vivek’s ability to work together with other researchers at international level and to contribute his strengths for a greater success of the team. (-- add exascale --)

Concurrently, Vivek worked on enhancing his strategy on using the measurement-based load balancing techniques used in Charm++ RTS to further guide his schedulers. This allowed performance gains on production machines such as ORNL's Cray and TACC's Ranger. This is a fundamental enhancement because it allows his scheduler using knowledge of the previous iterations to make scheduling decisions. Through the use of these techniques VIvek's scheduler incorporated an aspect of intelligence during the application's execution, which allowed for increasing performance of an application. These enhancements are particularly beneficial for bulk-synchronous simulations involviong several application timesteps. This work led to a publication of a paper in HiPC2011. This was done in collaboration with Abhinav Bhatele, at that time PhD student/postdoc working within the Charm++ group at Illinois, and now a post-doc at LLNL. This conference is highly competitive (~19% acceptance rate). Vivek presented this work at the conference in Bangalore, India in December 2011.

In January 2012, Vivek was awarded a Lawrence Scholar Fellowship, which awarded him funding from the United States Department of Energy to do research on his scheduling strategies for mitigating the amplification on DOE supercomputers. The award funding is for up to 4 years. He has worked from Lawrence Livermore Laboratory in California while doing his PhD at Illinois. His collaboration team involves Bronis de Supinski (mentor), Todd Gamblin, Torsten Hoefler, and Bill Gropp (advisor). The five of them meet for weekly conference calls, discussing ongoing progress on the implementation of his scheduling techniques. During his time at LLNL up to this point, Vivek has investigated how to make his scheduling approach usable for real-world applications. He has developed, under the supervision of Bronis de Supinski, a dynamic scheduler library integrated within OpenMP, where OpenMP uses information of the MPI runtime to possibly further guide the scheduling strategies. Specifically, each process tunes the scheduler parameters based on its properties. This work resulted in contributions of a source-to-source transformation of OpenMP codes, along with slack prediction techniques. He also addressed the problem of building an interface between two programming models within an hybrid programming model. He has developed other techniques such as using a scheduler variable task sizes, and integrated them into his software infrastructure. He along with his co-authors Bronis de Supinki, Todd Gamblin, Torsten Hoefler, and Bill Gropp has published a poster in the Supercomputing Conference in 2012 and a journal paper on dynamic scheduling techniques. They recently submitted to a top-tier high-performance computing conference the work on experimentation and development of schedulers done at LLNL, with a particular emphasis on the enhancements to the scheduler strategies done through systematic software implementation. With this, Vivek developed a scheduling library to attain additional performance benefits of pure static scheduling in his mixed static/dynamic technique, focusing on load imbalanced applications which use partitioning of work (e.g., Morton Orders for Molecular Dynamics Simulations) to improve spatial locality, which is beneficial only when using static scheduling. Vivek made this pre-determined partitioning of work still be beneficial in the context of his mixed static/dynamic scheduling strategy. This led to a paper in EuroMPI 2014 presented in Madrid, Spain in September.

Vivek also concluded his collaboration with Simplice in November when he presented a poster with Simplice at SC14 in New Orleans, in which he showed how different mixtures of scheduling strategies could be defined to handle different tradeoffs between load balance and locality. In this poster, he discussed a theoretical analysis of the work, and general scheduling strategy library for both numerical linear algebra and scientific applications.

The above elements and the global consistency of his research projects clearly demonstrate that Vivek Kale has the maturity of an excellent researcher, developing a timely and important research topic for the HPC community and exploring many of its theoretical and practical aspects, including the best way for others to use his approach.

Vivek defended his PhD last week, with his thesis dissertation focusing on low-overhead scheduling for improving performance of synchronous MPI programs. He is looking for postdoc positions to continue his research on low-overhead schedulers by integrating his schedulers with classical across-node load balancers, and to use techniques for any hybrid MPI+X models, not just MPI+OpenMP models.

In conclusion, Vivek Kale is an outstanding Ph. D. student. His research topic concerns a critical problem that needs to be solved in the near future. This is one of the fundamental problems in parallel computing and in the general field of computer science. This problem will be relevant as long as extreme-scale computing is necessary for the advancements in science and engineering. He already has established strong contacts and developed collaborations with several top researchers in the HPC domain. His work has shown to be directly beneficial and relevant in both a theoretical as well as practical setting. Based on Vivek's current work, along with his recently defined long-term goals as discussed in his letter of motivation, Vivek will benefit greatly through interactions with top scientists at the Heidelberg Laureate Forum, and also benefit from networking with other young aspiring computer scientists and mathematicians. Most importantly, I believe he will contribute to lively discussions and will go with motivation and great interest if he is invited to attend. Vivek has my highest recommendation to be a part of the Heidelberg Laureate Forum in August of this year.

What current problems within the areas of mathematics and computer science can my dissertation research provide benefit?

Longer-term goals (listen to what’s going on around you): Current state: HPC skills are important in a variety of fields. I am interested in continuing showing that my ideas are good, and developing my resume to be a leader in a company. I would like to make impacts for HPC in industry. I think a financial company is ideal.

Break down the goal: To do this, … I will need to … to finish the supporting data for nas lu and nbody. I will then send slides. I will then collect items from emails, and write them to organize them into the thesis. I plan to start writing the thesis chapter 1. I will give you updates periodically through the winter break.

Planning and logistics and politics … After I finish the thesis, I plan to contact others. I will also check with the academic office, and let them know. Note that the graduate college needs the thesis to be finished by … My hope is that there will be no other issues with the thesis that people have.

Future work: I then want to formalize the ideas. I hope that I can apply and formalize the ideas to other areas such as cloud computing. I then want to formalize the ideas to be societal. Media labs.

To do this:

1. need to apply to jobs.
2. Talk to people.

Other directions (if the above doesn’t work):

1. Go to google/linkedin. Keep the papers going. Justify for why this works: This will allow me to maintain strength in what I do.

2. Go to media lab. Justification for why this works.

***Work bios:***

***Recommendation letters:***

After the defense:

Strengthening: Building a network within work (think about how to do this, and how much priority this has): business industry, post-doc community, hpc community, tech industry.

Life: Build social networks

Find a girl:

- meet through matrimony:

- Building a network outside of work (think about how to do this, and how much priority this has): facebook, SF events, weddings, Chicago events,

- PC/mgmt: Become better at work-life.

C: 8 years : 40000

C++: 3 years : 12000

Python: 2 : 2500

Perl: 1: 500

VHDL: 0.5: 1000

Awards:

Lawrence Scholar Fellowship: This is a fellowship sponsored by the Lawrence Livermore National Laboratory and Department of Energy providing up to 4 years of funding for graduate research. Each year a selection from a pool of candidates consisting of PhD students is made from areas including Physics, Chemistry, Biology, and Computer Science. This fellowship award allows a PhD student to conduct their PhD research at Lawrence Livermore National Laboratory in California, with guidance for research given by their advisor and one or more mentors at Lawrence Livermore National Laboratory. This fellowship was awarded to me in November of 2011.

Notes on motivation and terminology: During execution of a scientific application on a supercomputer consisting of multi-core nodes, scientific application codes can significantly degrade in performance due to within-node load imbalances. My thesis work aims to address this through the use of low-overhead dynamic multi-core scheduling strategies, or ‘low-overhead scheduling’ for short. The basic technique of our low-overhead scheduling technique is described in reference 4 of the publications section of my CV.

My most significant accomplishment was obtaining on a multi-core node of a supercomputer a 64% performance improvement over a highly optimized (Communication-Avoiding) LU factorization multi-threaded implementation using low-overhead scheduling strategies along with a 34% performance gain over Intel’s MKL library implementation of LU factorization. The experimental results obtained were significant because LU factorization is a time-consuming numerical algorithm used in a large number of scientific application codes for engineering, such as the simulation of air flow over a plane’s wing, and so our scheduling strategy can help improve performance of those application codes. The results were also significant because they showed the potential of our scheduling strategies to improve performance of other scientific application codes. This work was published in 2012 at IPDPS, a top conference in the area of high-performance computing.

Note: The motivation for the development of each of the designs and terminology is explained in the most significant accomplishment (“Note on …”).

One design was a low-overhead scheduling library for improving performance of implementations of dense matrix factorization computations (for example, LU factorization) running on multi-core nodes of supercomputers. This size of the library was about 4000 lines of code. The code was written in C. I wrote roughly 50% of the code for the library, and the other 50% was written by my collaborator, Simplice Donfack of INRIA-Saclay. The purpose of this library was to demonstrate how dense matrix factorizations, a type of numerical algorithm frequently used in scientific application codes, could obtain performance improvements using low-overhead schedulers specifically designed for tuning performance of dense matrix factorization computations. This library also served to demonstrate how numerical linear algebra computations in general could obtain performance improvements through the use of these scheduling strategies (and the code of this library is being integrated into a large software system for numerical linear algebra computations called DPLASMA). The functionality of this library is as follows. Given a threaded dense matrix factorization code, a user adds our library’s scheduler functions before and after the threaded dense matrix factorization computation within the code. The user then experimentally determines the scheduler’s parameter value that would obtain the best performance for the code during its execution. These functions schedule work of the dense matrix factorization computation in a way that is low-overhead through the use of an intelligent blend of static and dynamic scheduling along with the use of experimental tuning of the scheduler parameters by the application programmer. The expected result of applying our library’s scheduling strategies to a dense matrix factorization code is a faster execution of that dense matrix factorization code on multi-core nodes.

Another design was a low-overhead scheduling library for scientific application codes. The size of the library is about 5500 lines of code. The library was written in C. My advisor led the project for the design of this library, and I wrote all of the code for the library. The purpose of the library was to provide our scheduling strategies generally for threaded scientific application codes, rather than specifically for threaded numerical linear algebra codes. The set of scheduling strategies used in this library are more sophisticated (though less special-purpose) than that of the library for dense matrix factorizations. Details of these scheduling strategies are in reference 6 in the publications section of the CV. The functionality is as follows. Given a threaded scientific application code implemented using MPI, an application programmer adds the library’s functions before and after the threaded computation regions of the application code. These functions are used by the runtime to schedule the work in the threaded computation region, using one of the scheduling strategies available in the library (with the application programmer getting to choose it). The user then experimentally determines the best performing scheduler’s parameter value(s). The expected result of this design is reduced application code execution time compared to that of the original application code.

A third design was a runtime software system for our scheduling strategy, which automatically adjusted the parameters of the low-overhead scheduling strategies implemented in the scheduling library described above. This size of the runtime software is about 7500 lines of code. The runtime is written in C++. Additionally, supporting application profiling was done using python. My advisor and LLNL supervisor led the project for this runtime. I wrote about 80% of the code for the runtime, and the other 20% was written in conjunction with software engineers at Lawrence Livermore National Laboratory. The purpose of the project was to provide a runtime to automatically adjust scheduler parameters for obtaining further performance improvements of application codes, and to substantially reduce application programmer effort to use our technique through automatic transformation of the application programmer’s code. The functionality is as follows. Given a threaded scientific application code using MPI, the application programmer compiles the code with our provided compiler. Our provided compiler modifies the code by placing our runtime’s function calls at the beginning and end of the threaded computation regions of the code. Then, the application programmer runs the code using our runtime software. On every invocation of the threaded computation region of the code, the function at the beginning of the region decides the best parameter values for the scheduling strategy applied in the immediately subsequent computation region. The parameter values are chosen based on history gathered through the function call invoked at the end of the computation region. The expected result is reduced application code execution time compared to that of the original application code.

Overview:

A runtime software system for our scheduling strategy, which automatically adjusted the parameters of the low-overhead scheduling strategies implemented in the scheduling library described above.

Purpose:

1. provide a runtime to automatically adjust scheduler parameters for obtaining further performance improvements of application codes.

2. substantially reduce application programmer effort to use our technique through automatic transformation of the application programmer’s code. Functionality:

**Spaces:**

***Jean Pockets:***

* Left pocket: Keys (RSA,car,house), cell phone (charger if needed) – unless it’s charging then remember to take it - Treat your phone like its your baby !!!
* Right pocket: wallet (40 dollars, credit card, driver's license, school id) - treat your wallet like its your baby 2 !!!

Vulnerability: right after coming home from car, taking off jeans for the night, at a club

***Backpack***:

* main compartment: computer case (computer, computer charger)
* side compartment: keys, checkbook, pens
* left side outer: usb charger, outlet
* right side outer: lotion
* right side inner: glasses
* left side inner: badge

Vulnerability: don’t put on floor of hotel or on bed/couch

***Car***:

* Glove compartment: insurance card
* Trunk: water, clif bars, shorts
* center console: earpiece, watch, sunglasses
* passenger underseat: maintenance papers
* driver underseat: checkbooks, receipts
* windshield wipers, windshield washer fluid
* tires check, oil change, maint req’d light
* gas, windshield cleaning
* Interior cleaning: front-inside of windshield, front seats, floor mats, side windows, center console, instrument panels, rear-view mirror, inside door entry
* Exterior cleaning: wheels, hood, front sides, front door panels, rear door panels, trunk+bumper, roof

Vulnerabilities: have a busy week of driving and forget to fill gas, use earpiece to talk on phone

***Office:***

* Left side: Fan, clock, phone, to file
* Right side: Food, tissue, papers, pens, hand sanitizer, tissue
* Cabinet 1: Right: work docs + books, Left: life docs + books,
* Center: Toiletries
* Cabinet 2: Papers
* Cabinet 3: Sleeping bag, pillow, misc., food
* Center: Computer
* Desktop:
* Docs:
* desktop background

Vulnerabilities: whenever I update the work docs,

***Room:***

***Bathroom***:

* clean toilet bowl
* check clothes in towel bar
* watch for water on floor
* clean sink area
* check for hairs on bathtub

Vulnerabilities: whenever done with shower

***Kitchen:***

* take garbage out
* clean /mop floor, clean countertops, watch for boiling water

Finish organization book, Julie Mortgenson’s organization

Doors locked when going out, garage door, check stove

***Appearance:***

* contacts / check for glasses
* make sure belt is on, clean shirt
* pant should be high

***Facebook***

***LinkedIn***

***Laptop***

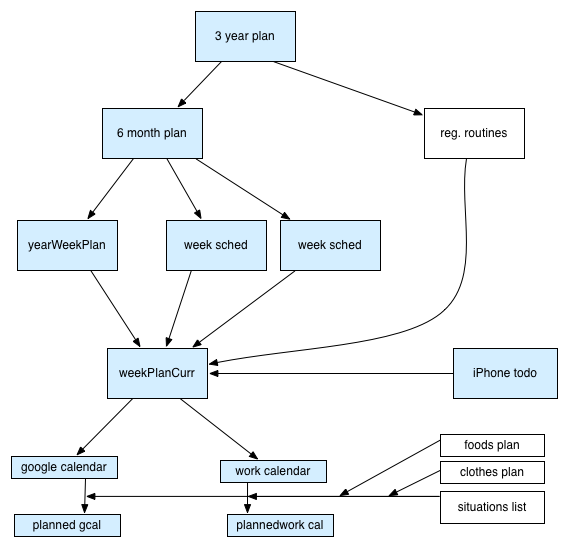
Make a daily plan to handle these things:

jean pockets (KeyWalletCell), Backpack (no crumbs, organize all pockets), office (clean desk of papers, get rid of hairs), laptop (clean outside, org folders), room (make bed, fold laundry, make sure no papers lying around), car (clean outside, get rid of crumbs, clean inside windshield), kitchen(milk in fridge, watch boil water), bath(water on floor, toilet bowl, take clothes out), lock doors, wash hands before/after eating something

When to throw things out:

Where to place certain things:

**Time:**



3 year plan:

Year week plan: Project goals list, Life goals list, monthly plan, Week-by-week plan

Project todo list:

iPhone todo list: use labels Mgmt, Comm, Rel., add on

Weekly Mgmt: Weekly to do (look at items from projects, from iPhone, from gchat), Making time Estimates, Prioritization, Schedule

Regular Routines:

*Every Morning:* Oral Hygiene (5 mins), Brush Teeth**,** Mouthwash**,** Clean tongue - it helps with breath **,** Eating small breakfast**,** Check your schedule for meetings / appointments for that day: Is anything urgent? 5 mins**,** Excercise: 1 hour (Stretch for 10 mins, Treadmill for 30 mins , Lift weights or Yoga 1 hour, Meditation for 10 mins)**,** Shave:5 mins (at least every other day) **,** Shower: 10 mins (shampoo+conditioner, acne wash,  soap on underarms/underbody) , dry hair, dry face, arms, legs, back, stomach, Clothing 5 mins to 10 mins**,** Comb hair, hair gel optional:(1 min.)**,** Remember to lock door, Keys/Wallet/Cell/Badge (get a bucket) (1 min.)**,** Any required medicines (1 min)**,** Wear lenses and rinse out case (2 mins)**,** Deodorant or cologne depending on the day 1 min**,** Wash your face with a face wash - it helps with acne (3 mins.)**,** Moisturize - apply lotion  (2 min)**,** Neti pot / clean out sinuses - if you’re feeling congested:5 mins **,** clean out ears, nose: 5 mins

*Every Evening / Night:* Call family: 30 mins**,** meet friends: 20 mins, Cleaning dishes / cleaning house: 10 mins, Lay out clothes for the next day, iron if necessary, start laundry if needed: 10 mins, Collect anything you’ll need for the next day (e.g. dry cleaning to drop off): 5 mins, Check Facebook/linkedin/google+: 10 mins, matrimony stuff: 10 mins, Food for lunch/dinner the next day: 10 mins, Situation analysis: 10 mins

*Going to Bed:* Finish computer stuff: 5 mins ***,*** Get into sleeping clothes: 1 mins, Before sleeping: brush teeth, floss - 2 mins ***,*** Properly wash and put away contacts: 2 mins***,*** Review the days good things: 10 mins ***,*** Light Reading: 10 mins, Prayer: 10 mins

Sat: analyze past week, spaces org, clean room,

Sun: long-term planning, week plan, running,

Mon: meeting notes, coding , prep for Toastmasters

Tuesday: Weights, Toastmasters @ 12PM

Wednesday: Weights

Thursday: Yoga

Friday: Weights

Clothes planning: coordination/Appearance details/ matching/ what to wear when: shirts, pants, underwear, socks, have combos set up, and put them on calendar

Foods planning: Groceries (milk, juice, bread, fruits, frozen foods), Dinner (spaghetti, indian, Mexican, Burger, Burrito)

Calendar:

* every year, put new routines on calendar
* every month, put upcoming big travel plans
* every week, update calendar
* every day, look at calendar for the next day, and see which situations arise.

Identify situations and put them up on calendar, and then put the situations pre-notes and post-notes to the calendar.

Think 20 mins ahead to see what’s happening.

Think a day ahead of time what’s happening and plan ahead .

Gen Knowledge: Directions/Logistics/Driving/Car

Gen. Knowledge: weather, technology, news.

Dinner Etiquette:

Health/Gym:

Budget/Money:

**Elements of Style:**

When to use semi-colon, put emphatic words at the end, that vs. which, gen. principles (write from the heart, have sympathy for reader)

**How to Win Friends and Influence People:**

- Never criticize, condemn or complain

- Give honest and sincere appreciation

- Arouse in the other person an eager want

**Conversation Skills**

(look at video)

**Emotional Intelligence + Social Intelligence**

(write down intro)

general evaluator

timer/grammarian

speaker

speech evaluator

table topics presenter, table topics answering

facebook, linkedin, Microsoft,

(Organize a bit better – 30 mins) :

***Matrimony/Relationship Mgmt :***

Picture1 , Picture 2

Later (R1438138) 

**A few words about my son:**

My son grew up in the US. He is a handsome and athletic person. He is hardworking and studious. In his free time, he enjoys playing piano. He also plays and follows tennis. He is doing a PhD in Computer Science, and has a fellowship to work in a government laboratory while doing it. **Basic Details:** Name: Later, Body Type / Complexion: Athletic / Wheatish, Age: 29 Yrs, Physical Status: Normal, Height: 6 Ft / 183 Cms, Weight: 76 Kgs / 167 lbs, Mother Tongue: Marathi, Marital Status: Never married, Eating Habits: Eggetarian (vegetarian), Drinking Habits: Light / Social drinker, Smoking Habits: Non-smoker **Religious Information:** Religion: Hindu, Caste / Sub Caste: Brahmin Deshastha (Caste No Bar) / Rigvedi, Gothram: Vishwamitra, Star / Raasi: Not Specified, Manglik: Don't know **Location:** Country: United States of America, City: Berkeley, State: California, Resident Status: Citizen, Citizenship: United States of America **Professional Information:** Education: MSc IT / Computer Science, Education in Detail: BS Computer Science, then worked for a while before returning to higher studies, Occupation: Student, Occupation in Detail: PhD student/ Fellowship, Employed in: Government/PSU, Annual Income: Not Specified **Hobbies & Interests:** Hobbies: playing piano, Interests: Politics, Health & fitness, Favorite Music: Hip-Hop, Rap, Techno, Jazz, Western classical, Pop, Sports/Fitness Activities: Cycling, Tennis, Jogging / walking, Weight lifting, soccer, Favorite Cuisine: South Indian, Thai, Punjabi, Italian, Preferred Dress Style: Casual wear, Western formal wear, Spoken Languages: English, Marathi

**Family Details:** Family Values: Moderate, Father's Status: Professor, Family Type: Nuclear, Mother's Status: homemaker, Family Status: Upper middle class, No of Brother(s): 1 - Not Married, Ancestral Origin: Not Specified, No of Sister(s): None **About our family:** We are a close-knit family. We have been in the US for the last 30 years. His younger brother works in Chicago. We are well-connected with India, with siblings in Pune.

**PARTNER PREFERENCE**

**Basic & Religious Preferences**

Brides's Age: 24 – 30 Yrs, Height: 5 Ft - 6 Ft / 152 Cms - 183 Cms , Physical Status: Normal, Eating Habits: Doesn't matter, Smoking Habits: Non-smoker, Drinking Habits: Doesn't matter, Religion: Hindu, Mother Tongue: Any Mother Tongue, Caste: Any Caste, Sub Caste: Any Sub Caste, Gothram: All (Except my gothra), Star: Any Star, Manglik: Doesn't matter**,** Education: Any Degree, Occupation: Any Occupation, Annual Income: Any Income**,** Citizenship: Any Citizenship, Country: United States of America, (India), Residing State: Any, Any, Residing City: Any City

**What we are looking for**

|  |
| --- |
| Must be willing to live in the US. Prefer someone who spent many years in the USA.  We hope to find someone who has a happy disposition and understanding nature. |

*Rules for parents involvement:*

1. Every on Sunday, both Vivek and mom and dad go through the people and filter those that are likely candidates.
2. Have a call with me to discuss “yes”, “no”, “maybe”. For those that are “yes”, respond within two weeks. For those that are “no”, respond giving one of the stock responses(“looking for U.S. citizens” , “not what we are looking for”).
3. If pictures are asked for, give only 3 additional pictures. These should be the same for each person that requests.
4. In the initial interest by parents, do not try to sell yourself and me at all. Just answer the questions that are asked. They can already google things about you to figure out whether you are a “Professor at the University of Illinois”.
5. When the mother is talking, please have mom talk to her directly. I don’t want dad talking to the mother.
6. Also, I want mom’s signature to be on some of the emails.
7. When asked about specific information about me, such as “what I’m doing , plans for the future”, say vehemently that “Vivek will tell more about this”
8. Before getting involved in a phone call, please let me know about it!
9. Before starting an email exchange (even if they need a response quickly), please let me know about it! Let me know about specific questions you might have.

*Stock Responses:*

*Interests:*

*Mgmt:*

- Every weekend, set aside some time for responding.

- Every night, check new interests and add on to list.

- talk to parents

Dating guide:

* Don’t talk about past relationships
* Know when to hold hands, when to show affection, when to kiss
* Listen carefully to conversation
* Don’t say anything negative about girl, even if she says negative to you.
* Stay calm in tough situations
* Be happy and smile, act like you are having fun
* Add humour every now and then
* Connecting/find common ground
* Understand married life
* Send flowers to show you care, when interested.
* Be intentional with what you are saying, make sure you organize your thoughts
* Shout to look confident
* Avoid talking too long on phone, keep dates short and simple
* Remember that a girl’s mind can’t forget things often

Channels :

* Match
* Matrimony
* Eharmony
* CoffeeMeetsBagel