**Answer the first five questions (A-E) on this paper. Submission instructions are at the end of the exam. To earn 2/2 marks on a question, you must give a very thoughtful, intentional answer. These *are* worth marks.**

**Name: \_\_\_\_\_\_\_\_\_\_Lei Pei\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Marks: \_\_\_\_\_\_\_\_ / 35**

**Student Number: \_\_\_\_\_ A00924846\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question A 2 marks**

**Question B 2 marks**

**Question C 2 marks**

**Question D 2 marks**

**Question E 2 marks**

**Question F 25 marks**

**Read the following five articles and answer two questions for each.**

**Answer in a way that shows what is most important TO YOU, not to the author or your instructor.**

**READ and ANSWER TWO QUESTIONS PER ARTICLE**

# QUESTION A: From Inc.com Markus Frind and PlentyOfFish.com:

**Markus Frind recently increased his staff from zero to three and moved operations from his bedroom to a nearby office tower.**

**At 10 o'clock in the morning, Markus Frind leaves his apartment and heads to work.**

**It's a short walk through downtown Vancouver, British Columbia, but somehow the trek feels arduous. This is n**

**ot because Frind is lazy. Well, Frind is a bit lazy, but that's another matter. The problem is that he is still getting used to the idea of a commute that involves traveling farther than the distance between the living room and the bedroom.**

**Frind's online dating company, Plenty of Fish, is newly located on the 26th floor of a downtown skyscraper with a revolving restaurant on the roof. The gleaming space could easily house 30 employees, but as Frind strides in, it is eerily quiet -- just a room with new carpets, freshly painted walls, and eight flat-screen computer monitors. Frind drops his bag and plops himself down in front of one of them.**

**He looks down at his desk. There's a $180,000 order waiting for his signature. It's from VideoEgg, a San Francisco company that is paying Frind to run a series of Budweiser commercials in Canada. Like most of his advertising deals, this one found Frind. He hadn't even heard of VideoEgg until a week ago. But then, you tend to attract advertisers' attention when you are serving up 1.6 billion webpages each month.**

**That's a lot of personal ads. "One-point-six ba-hillion," Frind says slowly, smacking his lips on the hard b. "There are maybe 10 sites in the U.S. with more than that." Five years ago, he started Plenty of Fish with no money, no plan, and scant knowledge of how to build a Web business. Today, according to the research firm Hitwise, his creation is the largest dating website in the U.S. and quite possibly the world. Its traffic is four times that of the dating pioneer Match, which has annual revenue of $350 million and a staff that numbers in the hundreds. Until 2007, Frind had a staff of exactly zero. Today, he employs just three customer service workers, who check for spam and delete nude images from the Plenty of Fish website while Frind handles everything else.**

**Amazingly, Frind has set up his company so that doing everything else amounts to doing almost nothing at all. "I usually accomplish everything in the first hour," he says, before pausing for a moment to think this over. "Actually, in the first 10 or 15 minutes."**

**To demonstrate, Frind turns to his computer and begins fiddling with a free software program that he uses to manage his advertising inventory. While he is doing this, he carps about Canada's high income-tax rate, a serious problem considering that Plenty of Fish is on track to book revenue of $10 million for 2008, with profit margins in excess of 50 percent. Then, six minutes and 38 seconds after beginning his workday, Frind closes his Web browser and announces, "All done."**

**All done? Are you serious? "The site pretty much runs itself," he explains. "Most of the time, I just sit on my ass and watch it." There's so little to do that he and his girlfriend, Annie Kanciar, spent the better part of last summer sunning themselves on the French Riviera. Frind would log on at night, spend a minute or two making sure there were no serious error messages, and then go back to sipping expensive wine. A year ago, they relaxed for a couple of weeks in Mexico with a yacht, a captain, and four of Kanciar's friends. "Me and five girls," he says. "Rough life."**

**As Frind gets up to leave, I ask him what he has planned for the rest of the day. "I don't know," he says. "Maybe I'll take a nap."**

**It's a 21st-century fairy tale: A young man starts a website in his spare time. This person is unknown and undistinguished. He hasn't gone to MIT, Stanford, or any other four-year college for that matter, yet he is deceptively brilliant. He has been bouncing, aimlessly, from job to job, but he is secretly ambitious. He builds his company by himself and from his apartment. In most stories, this is where the hard work begins -- the long hours, sleepless nights, and near-death business experiences. But this one is way more mellow. Frind takes it easy, working no more than 20 hours a week during the busiest times and usually no more than 10. Five years later, he is running one of the largest websites on the planet and paying himself more than $5 million a year.**

**Frind, 30, doesn't seem like the sort of fellow who would run a market-leading anything. Quiet, soft-featured, and ordinary looking, he is the kind of person who can get lost in a roomful of people and who seems to take up less space than his large frame would suggest. Those who know Frind describe him as introverted, smart, and a little awkward. "Markus is one of those engineers who is just more comfortable sitting in front of a computer than he is talking to someone face to face," says Noel Biderman, the co-founder of Avid Life Media, a Toronto-based company that owns several dating sites.**

**When he does engage in conversation, Frind can be disarmingly frank, delivering vitriolic quips with a self-assured cheerfulness that feels almost mean. Yahoo, he says, is "a complete joke," Google is "a cult," and Match is "dying." Says Mark Brooks, a marketing consultant who has advised Frind since 2006, "I've never known anybody so competitive. He always says exactly what he thinks."**

**With friends and family, Frind expresses affection through playful pranks. Frind will spend hours hiding in the three-bedroom apartment he and Kanciar share, furtively flipping light switches, tapping on doors, and ducking into rooms to play on his girlfriend's fear of ghosts. Another memorable valentine involved the secret consumption of a massive quantity of hot peppers. Though his mouth was on fire, Frind calmly planted a kiss on Kanciar's lips and feigned ignorance as she went scrambling for water.**

**Kanciar, a freelance Web designer who also helps out around Plenty of Fish, is a lanky blonde with an easy smile and a hearty laugh, which she uses liberally to try to get Frind to open up. When I ask him to talk about what he does with the 23 hours a day in which he doesn't work, Frind struggles to answer and then looks helplessly at Kanciar. She offers a few suggestions -- video games, ski trips, walks -- and then tries to focus his energies. "We're trying to convince Max that we're interesting," she says sweetly.**

**That's not easy for Frind, who seems most comfortable with the world at arm's length. "He never raises his voice," Kanciar says later. "And he doesn't like conflict." Frind prefers to remain a silent observer of others who then constructs arguments and counterarguments about their motivations. He seems perpetually lost in thought, constantly thinking about and studying the world around him. "He's always watching his environment to apply it to the site," says Kanciar. "Once in a while, from the middle of nowhere, he'll say, 'Why is that girl doing that?' or 'Why is that guy posing like that?' He'll check people out in restaurants and watch how they interact. In a way, he's thinking about the company all the time."**

**Frind spent his formative years on a grain farm in the northern hinterlands of British Columbia -- "the bush" in local parlance. His hometown, Hudson's Hope, is a cold, isolated place not far from the starting point of the Alaska Highway. Frind's parents, German farmers who emigrated just before his fourth birthday, bought a 1,200-acre plot 10 miles from town and initially lived in a trailer without electricity, phones, or running water. The family's closest neighbors were a mile and a half away, and, apart from a younger brother, Frind had few friends. "His problem was English," says his father, Eduard Frind. "If you don't have English, you can't do anything." Frind eventually adjusted, but his was a lonely childhood. He rarely visits Hudson's Hope these days. When his parents want to see him, they make the 14-hour drive southward.**

**After graduating from a technical school in 1999 with a two-year degree in computer programming, Frind got a job at an online shopping mall. Then, the dot-com bubble burst, and he spent the next two years bouncing from failed start-up to failing start-up. For most of 2002, he was unemployed. "Every six months, I got a new job," Frind says. "It'd start with 30 people, and then five months later, there'd be five. It was brutal." When he did have work, it felt like torture. His fellow engineers seemed to be writing deliberately inscrutable code in order to protect their jobs. "It would literally take me four or five hours," he says -- an eternity in Markus Frind time -- "just to make heads or tails of their code, when normally you're supposed to spend, like, two minutes doing that."**

**But cleaning up other people's messes taught Frind how to quickly simplify complex code. In his spare time, he started working on a piece of software that was designed to find prime numbers in arithmetic progression. The topic, a perennial challenge in mathematics because it requires lots of computing power, had been discussed in one of his classes, and Frind thought it would be a fun way to learn how to sharpen his skills. He finished the hobby project in 2002, and, two years later, his program discovered a string of 23 prime numbers, the longest ever. (Frind's record has since been surpassed, but not before it was cited by UCLA mathematician and Fields Medal winner Terence Tao.) "It was just a way of teaching myself something," Frind says. "I was learning how to make the computer as fast as possible."**

**By early 2003, the technology economy in Vancouver had yet to bounce back, and Frind's sixth employer in three years was laying off half its work force. Worried that he would again find himself unemployed, Frind decided he needed to bolster his qualifications. He would devote a couple of weeks to mastering Microsoft's new tool for building websites, ASP.net, and he would do it by building the hardest kind of website he could think of.**

**Online dating was an inspired choice. Not only does the act of building an intricate web of electronic winks, smiles, and nudges require significant programming skills, but the industry has always been a friendly place for oddballs and opportunists. Industry pioneer Gary Kremen, the founder of Match and the man who registered the Sex.com domain name, cites rapper Ice Cube and the bank robber "Slick" Willie Sutton as important influences on his business philosophy. Another pioneer, James Hong, co-founded Hot or Not, a site with a single, crude feature. Hong allowed users to upload pictures of themselves and have other users rate their attractiveness on a scale of 1 to 10. Hot or Not was acquired for $20 million in cash last year by Noel Biderman's company, Avid Life. Avid, which has also courted Plenty of Fish, derives most of its revenue from Ashley Madison, a dating website for married people (tag line: "Life is short. Have an affair"). The site has 2.8 million members and revenue in the tens of millions of dollars.**

**Unlike many online dating entrepreneurs, Frind didn't start Plenty of Fish to meet women -- or even because he had some vision of business glory. "It was a burning desire to have something stable," he says. "And I didn't really want to work." Frind's eyes were also a factor. He suffers from hypersensitivity to light, and his eyes were not taking well to long days in front of a screen. Working a few hours an evening for two weeks, Frind built a crude dating site, which he named Plenty of Fish. It was desperately simple -- just an unadorned list of plain-text personals ads. But it promised something that no big dating company offered: free.**

**The idea had initially come to Frind in 2001, when he started checking out Canada's then-largest dating site, Lavalife, hoping to meet women or at least to kill some time. Online dating seemed like a good idea, but he was startled to discover that the site charged users hefty fees. "I thought it was ridiculous," he says. "It was this rinky-dink little site charging money for something anyone could make. I was like, I can beat these guys."**

**This thought was not exactly new. Since the mid-'90s, there had been dozens of free dating start-ups, but all had struggled to attract users, because they were competing with the outsize marketing budgets of paid competitors like Lavalife. Paid sites could afford to spend $30 or $40 in advertising to acquire a user. A free site could afford to spend perhaps 40 cents, making it exceedingly hard to attract daters and still turn a profit. Frind's answer to this problem was somewhat radical. Rather than try to compete directly with Match, the industry leader, he created a website that cost almost nothing to run and was aimed at the sort of people who wanted to browse a few profiles but weren't ready to take out their credit cards. In doing so, he had found a way to reach a large, underserved market. Even better, he had created a perfect place for paid dating sites to spend their huge advertising budgets.**

**Plenty of Fish grew slowly at first as Frind focused on learning the programming language and trolling Internet forums for clues on how to increase his traffic. There are a handful of half-literate posts from early 2003 in which Frind asks basic questions, like "I am interested in know how much money sites generate off advertising." Reading these comments in retrospect paints a picture of determination and naiveté.**

**Frind knew little about search-engine optimization or online advertising, but he was a quick study. From March to November 2003, his site expanded from 40 members to 10,000. Frind used his home computer as a Web server -- an unusual but cost-effective choice -- and spent his time trying to game Google with the tricks he picked up on the forums. In July, Google introduced a free tool called AdSense, which allowed small companies to automatically sell advertisements and display them on their websites. Frind made just $5 in his first month, but by the end of the year, he was making more than $3,300 a month, largely by selling ads to paid dating sites that were interested in getting his unpaid members to trade up. He quit his job.**

**"Have you ever met anyone like me?" This is both a boast and a genuine question: Frind has few friends in business, no mentors, and no investors. Moreover, he has taken a path that seems at odds with the conventional wisdom about Internet companies. Most websites with as much traffic as Plenty of Fish would have by this point raised millions of dollars from venture capitalists, hired dozens of engineers and business-development types, and figured out a way to keep someone as unconventional as Markus Frind from making any major decisions.**

**But if Frind's methods make him unusual, he is also a man of his times. In the past few years, a new technological ecosystem, built around Google's dominance in Web search and its decision to offer powerful software tools at no charge, has changed the economics of doing business on the Internet. Web analytic services that used to cost thousands of dollars a year are now free. Competitive data, once available to only the largest companies, can be had with only a few clicks on Compete.com and Quantcast.com. And advertising networks, especially AdSense, have made it possible, even preferable, for Internet entrepreneurs to bootstrap their businesses without hiring a sales force and without raising much money. Websites that venture capitalists would have spent tens of millions of dollars building in 1998 can now be started with tens of dollars.**

**No one has used this ecosystem as effectively as Markus Frind, who has stayed simple, cheap, and lean even as his revenue and profits have grown well beyond those of a typical one-person company. Plenty of Fish is a designer's nightmare; at once minimalist and inelegant, it looks like something your nephew could have made in an afternoon. There's the color scheme that seems cribbed from a high school yearbook and the curious fondness for bold text and CAPITAL LETTERS. When searching for a prospective mate, one is inundated with pictures that are not cropped or properly resized. Instead, headshots are either comically squished or creepily elongated, a carnivalesque effect that makes it difficult to quickly size up potential mates.**

**Frind is aware of his site's flaws but isn't eager to fix them. "There's no point in making trivial adjustments," he says. Frind's approach -- and the reason he spends so little time actually working -- is to do no harm. This has two virtues: First, you can't waste money if you are not doing anything. And second, on a site this big and this complex, it is impossible to predict how even the smallest changes might affect the bottom line. Fixing the wonky images, for instance, might actually hurt Plenty of Fish. Right now, users are compelled to click on people's profiles in order to get to the next screen and view proper headshots. That causes people to view more profiles and allows Frind, who gets paid by the page view, to serve more ads. "The site works," he says. "Why should I change what works?"**

**Frind has resisted adding other commonly requested features, such as chatrooms and video profiles, on the same grounds. "I don't listen to the users," he says. "The people who suggest things are the vocal minority who have stupid ideas that only apply to their little niches." Instead, Frind has focused his energy on making the site better at matching people. When a member starts browsing through profiles, the site records his or her preferences and then narrows down its 10 million users to a more manageable group of potential mates. "Users never see the whole database," Frind says. "It gets smaller and more focused on what you're actually looking for." In other words, if you tell Plenty of Fish you want to date blond nonsmokers but spend all your time gawking at nicotine-addled brunettes, the program will adjust. "People think they know who the perfect person is, but that's not always who they really want," he says. Frind estimates, based on exit surveys, that the site creates 800,000 successful relationships a year.**

**But the brilliance of Plenty of Fish is not its strength as a matching engine; it is the site's low overhead. Not only has Frind managed to run his company with almost no staff, but he has also been able to run a massive database with almost no computer hardware. To get a sense of how efficient the operation is, consider that the social news site Digg generates about 250 million page views each month, or roughly one-sixth of Plenty of Fish's monthly traffic, and employs 80 people. Most websites as busy as Frind's use hundreds of servers. Frind has just eight. He is not eager to explain how he manages this, but he says that it mostly comes from writing efficient code, a necessity when you are the only code writer and are extremely averse to spending money on additional hardware and features. "At other sites, when one thing goes slightly wrong, the reaction is to buy more servers or hire a Ph.D.," he says. "It's almost unbelievable -- it's like people are trying to justify their jobs by spending money. This isn't rocket science."**

**Often, at the end of a long workday, which is to say around noon, Frind plays war games. His apartment is outfitted with five computers for group play of Age of Empires and Command & Conquer -- and he has a substantial collection of board games. He is good, too: When I joined him for a game of Risk in October, he sat silently for almost the entire game before clearing the board in a single, virtuosic turn. He was still gloating the next morning. Frind approaches business in much the same way. "It's a strategy game," he says. "You're trying to take over the world, one country at a time."**

**Frind's account of his own exploits, published on his blog in 2006 under the title "How I Started a Dating Empire," says a lot about his worldview: "I spent every waking minute when I wasn't at my day job reading, studying, and learning. I picked out 'enemies' and did everything I could to defeat them, which meant being bigger than them. I refused to accept defeat of any kind." Around the same time, he returned to one of his old Internet hangouts, a forum called WebmasterWorld, and posted a brief how-to guide entitled "How I Made a Million in Three Months." It contained a blueprint for the success of Plenty of Fish: Pick a market in which the competition charges money for its service, build a lean operation with a "dead simple" free website, and pay for it using Google AdSense.**

**By 2006, Plenty of Fish was serving 200 million pages each month, putting it in fifth place in the United States and first in Canada among dating sites. Frind was making amazingly good money, too: $10,000 a day through AdSense. In March of that year, Frind mentioned these facts to Robert Scoble, a popular tech blogger whom he met at a conference in Vancouver. When Scoble wrote about the solo entrepreneur with the ugly website making millions of dollars a year, his readers were in disbelief. At the time, AdSense was seen as a tool for amateurs. It might cover your blogging expenses, but it wouldn't make you rich. Frind's website was also downright ugly. A search-engine-optimization blogger, Jeremy Schoemaker, wrote that Frind was a liar. "Give me a break, dudes," he wrote. "You look so stupid when you buy into his crap."**

**Frind embraced the controversy. He posted a picture of a check from Google for nearly a million Canadian dollars (or about $800,000) made out to Plenty of Fish. It represented two months' worth of revenue and implied that his site was making $4.8 million a year. But some thought the check was a fake, while others felt that posting it was a crude promotional stunt. "He came out of nowhere, and he didn't seem to give a shit," says David Evans, who writes the blog Online Dating Insider. But the stunt worked. Frind's site was the talk of the blogosphere, driving gobs of new users to the site. Plenty of Fish's growth accelerated dramatically, hitting one billion page views a month by 2007.**

**By the summer of 2008, with his site moving into first place among dating sites in the U.S. and the U.K., Frind began to wonder about his next step. He rented a 3,700-square-foot suite in Vancouver's Harbour Center, announced he was going to hire 30 employees, and bought a BlackBerry. But the plans were not exactly concrete. By October, Frind's own office was still empty: no furniture, nothing on the walls. He still hadn't figured out how to get e-mail on his cell phone. He had hired three people, not 30.**

**Frind seems untroubled by this disconnect. He says he leased an office because he was tired of working at home. He assumes he will one day need more employees, but he hasn't figured out what he would do with them. And he is in no hurry. He hasn't even bothered to offer a French language site for the six million French speakers living in Quebec. "I'll get around to doing that eventually," he says.**

**With all the free time on his hands, why doesn't Frind just start a second company? He says he thinks about that sometimes and has even toyed with creating a free job- listings site but finds the idea stultifying. "Sure, I could do it, but it'd be like watching grass grow compared to this," he says, gesturing at a traffic chart that shows Plenty of Fish's growth over the past few months. "It'd be like" -- he adopts a high-pitched, mocking tone -- " 'Whoop-de-doo, we got 100 visitors today.' Whereas with my site, every day there's another 1,000 or 10,000."**

**It's hard to know what to make of a guy who works an hour a day, who doesn't travel much, and who doesn't have any hobbies beyond war games somehow fretting about boredom. How is he not bored already?**

**But if Frind is guilty of a kind of sloth, there is also a wisdom to his passivity. Being ever careful takes serious self-discipline, and an aversion to doing harm can be more valuable than an overeagerness for self-improvement. If nothing else, it's impossible to argue with his success. Frind created his own game and wrote his own rules. As growing legions of lovesick people around the globe search for their perfect mates and advertisers fall over one another to write him ever larger checks, he just kicks back and smiles. And the money rolls in.**

**1a) Summarize the Markus Frind article in one short paragraph.**

**The idea that Frind wants to share with us is that making money is not totally based on working hard, it is more, however, based on working smartly. For most of the IT business, you don’t need to get big invest in order to build something, but this whole business has to offer what the market needs. If you build something that people don’t want to use or charge too much money on users, your business might fail too. Besides, it is sometimes not a good idea to build a second company if you already have a successful business working. You might as well think how to get the current one to work better.**

**1b) What (if anything) can you apply from this to your life/future/business?**

**Well, I like doing business. I actually created my own dating website when I was in my university in China. But I could not bring as many users as possible. So I failed and ended up selling it for only $3000. Right now I am trying to do business in Canada as well. What I learn from this article and my previous experience is that we should not , as a web-based company , always charge our users. I believe there are tons of users who hate being charged. Therefore, that is why traffic is the key for the success of a website. You need attract as many users as possible, so making it free to use is actually a good business strategy as well. Then you can make money off ads. Another thing I think I can what I can apply to my business is that we should not always fear competition. It is so smart if you can turn your business competitor to your business partners or even charge them.**

# QUESTION B: Paul Graham (startup incubator and venture capitalist):

**I realized recently that what one thinks about in the shower in the morning is more important than I'd thought. I knew it was a good time to have ideas. Now I'd go further: now I'd say it's hard to do a really good job on anything you don't think about in the shower.**

**Everyone who's worked on difficult problems is probably familiar with the phenomenon of working hard to figure something out, failing, and then suddenly seeing the answer a bit later while doing something else. There's a kind of thinking you do without trying to. I'm increasingly convinced this type of thinking is not merely helpful in solving hard problems, but necessary. The tricky part is, you can only control it indirectly.**

**I think most people have one top idea in their mind at any given time. That's the idea their thoughts will drift toward when they're allowed to drift freely. And this idea will thus tend to get all the benefit of that type of thinking, while others are starved of it. Which means it's a disaster to let the wrong idea become the top one in your mind.**

**What made this clear to me was having an idea I didn't want as the top one in my mind for two long stretches.**

**I'd noticed startups got way less done when they started raising money, but it was not till we ourselves raised money that I understood why. The problem is not the actual time it takes to meet with investors. The problem is that once you start raising money, raising money becomes the top idea in your mind. That becomes what you think about when you take a shower in the morning. And that means other questions aren't.**

**I'd hated raising money when I was running Viaweb, but I'd forgotten why I hated it so much. When we raised money for Y Combinator, I remembered. Money matters are particularly likely to become the top idea in your mind. The reason is that they have to be. It's hard to get money. It's not the sort of thing that happens by default. It's not going to happen unless you let it become the thing you think about in the shower. And then you'll make little progress on anything else you'd rather be working on.**

**(I hear similar complaints from friends who are professors. Professors nowadays seem to have become professional fundraisers who do a little research on the side. It may be time to fix that.)**

**The reason this struck me so forcibly is that for most of the preceding 10 years I'd been able to think about what I wanted. So the contrast when I couldn't was sharp. But I don't think this problem is unique to me, because just about every startup I've seen grinds to a halt when they start raising money—or talking to acquirers.**

**You can't directly control where your thoughts drift. If you're controlling them, they're not drifting. But you can control them indirectly, by controlling what situations you let yourself get into. That has been the lesson for me: be careful what you let become critical to you. Try to get yourself into situations where the most urgent problems are ones you want to think about.**

**You don't have complete control, of course. An emergency could push other thoughts out of your head. But barring emergencies you have a good deal of indirect control over what becomes the top idea in your mind.**

**I've found there are two types of thoughts especially worth avoiding—thoughts like the Nile Perch in the way they push out more interesting ideas. One I've already mentioned: thoughts about money. Getting money is almost by definition an attention sink. The other is disputes. These too are engaging in the wrong way: they have the same velcro-like shape as genuinely interesting ideas, but without the substance. So avoid disputes if you want to get real work done.**

**Even Newton fell into this trap. After publishing his theory of colors in 1672 he found himself distracted by disputes for years, finally concluding that the only solution was to stop publishing:**

**I see I have made myself a slave to Philosophy, but if I get free of Mr Linus's business I will resolutely bid adew to it eternally, excepting what I do for my privat satisfaction or leave to come out after me. For I see a man must either resolve to put out nothing new or become a slave to defend it.**

**Linus and his students at Liege were among the more tenacious critics. Newton's biographer Westfall seems to feel he was overreacting:**

**Recall that at the time he wrote, Newton's "slavery" consisted of five replies to Liege, totalling fourteen printed pages, over the course of a year.**

**I'm more sympathetic to Newton. The problem was not the 14 pages, but the pain of having this stupid controversy constantly reintroduced as the top idea in a mind that wanted so eagerly to think about other things.**

**Turning the other cheek turns out to have selfish advantages. Someone who does you an injury hurts you twice: first by the injury itself, and second by taking up your time afterward thinking about it. If you learn to ignore injuries you can at least avoid the second half. I've found I can to some extent avoid thinking about nasty things people have done to me by telling myself: this doesn't deserve space in my head. I'm always delighted to find I've forgotten the details of disputes, because that means I hadn't been thinking about them. My wife thinks I'm more forgiving than she is, but my motives are purely selfish.**

**I suspect a lot of people aren't sure what's the top idea in their mind at any given time. I'm often mistaken about it. I tend to think it's the idea I'd want to be the top one, rather than the one that is. But it's easy to figure this out: just take a shower. What topic do your thoughts keep returning to? If it's not what you want to be thinking about, you may want to change something.**

1. **Summarize the Paul Graham article in one short paragraph.**

**Paul is overwhelmed with stress on raising capital for his business, as well as having difficulty managing his personal life because he is work focus. Paul tries to convey the message that it is important to stay focus on your thoughts, brain storming ideas , and let your thoughts “drift” so to speak. But it is difficult to manage both work and personal life, if his life is about work.**

**b) What (if anything) can you apply from this to your life/future/business?**

**It is important to separate business, and personal life. Otherwise you end up like Paul, always “drifting” in thoughts. Another thing I can apply to my life is that our brain is always filled with many things and we should tell what are important and what are not. Then you should do those most important at first.**

# QUESTION C: Another Paul Graham essay:

**One of the things I always tell startups is a principle I learned from Paul Buchheit: it's better to make a few people really happy than to make a lot of people semi-happy. I was saying recently to a reporter that if I could only tell startups 10 things, this would be one of them. Then I thought: what would the other 9 be?**

**When I made the list there turned out to be 13:**

**1. Pick good cofounders.**

**Cofounders are for a startup what location is for real estate. You can change anything about a house except where it is. In a startup you can change your idea easily, but changing your cofounders is hard. And the success of a startup is almost always a function of its founders.**

**2. Launch fast.**

**The reason to launch fast is not so much that it's critical to get your product to market early, but that you haven't really started working on it till you've launched. Launching teaches you what you should have been building. Till you know that you're wasting your time. So the main value of whatever you launch with is as a pretext for engaging users.**

**3. Let your idea evolve.**

**This is the second half of launching fast. Launch fast and iterate. It's a big mistake to treat a startup as if it were merely a matter of implementing some brilliant initial idea. As in an essay, most of the ideas appear in the implementing.**

**4. Understand your users.**

**You can envision the wealth created by a startup as a rectangle, where one side is the number of users and the other is how much you improve their lives. The second dimension is the one you have most control over. And indeed, the growth in the first will be driven by how well you do in the second. As in science, the hard part is not answering questions but asking them: the hard part is seeing something new that users lack. The better you understand them the better the odds of doing that. That's why so many successful startups make something the founders needed.**

**5. Better to make a few users love you than a lot ambivalent.**

**Ideally you want to make large numbers of users love you, but you can't expect to hit that right away. Initially you have to choose between satisfying all the needs of a subset of potential users, or satisfying a subset of the needs of all potential users. Take the first. It's easier to expand userwise than satisfactionwise. And perhaps more importantly, it's harder to lie to yourself. If you think you're 85% of the way to a great product, how do you know it's not 70%? Or 10%? Whereas it's easy to know how many users you have.**

**6. Offer surprisingly good customer service.**

**Customers are used to being maltreated. Most of the companies they deal with are quasi-monopolies that get away with atrocious customer service. Your own ideas about what's possible have been unconsciously lowered by such experiences. Try making your customer service not merely good, but surprisingly good. Go out of your way to make people happy. They'll be overwhelmed; you'll see. In the earliest stages of a startup, it pays to offer customer service on a level that wouldn't scale, because it's a way of learning about your users.**

**7. You make what you measure.**

**I learned this one from Joe Kraus. Merely measuring something has an uncanny tendency to improve it. If you want to make your user numbers go up, put a big piece of paper on your wall and every day plot the number of users. You'll be delighted when it goes up and disappointed when it goes down. Pretty soon you'll start noticing what makes the number go up, and you'll start to do more of that. Corollary: be careful what you measure.**

**8. Spend little.**

**I can't emphasize enough how important it is for a startup to be cheap. Most startups fail before they make something people want, and the most common form of failure is running out of money. So being cheap is (almost) interchangeable with iterating rapidly. [4] But it's more than that. A culture of cheapness keeps companies young in something like the way exercise keeps people young.**

**9. Get ramen profitable.**

**"Ramen profitable" means a startup makes just enough to pay the founders' living expenses. It's not rapid prototyping for business models (though it can be), but more a way of hacking the investment process. Once you cross over into ramen profitable, it completely changes your relationship with investors. It's also great for morale.**

**10. Avoid distractions.**

**Nothing kills startups like distractions. The worst type are those that pay money: day jobs, consulting, profitable side-projects. The startup may have more long-term potential, but you'll always interrupt working on it to answer calls from people paying you now. Paradoxically, fundraising is this type of distraction, so try to minimize that too.**

**11. Don't get demoralized.**

**Though the immediate cause of death in a startup tends to be running out of money, the underlying cause is usually lack of focus. Either the company is run by stupid people (which can't be fixed with advice) or the people are smart but got demoralized. Starting a startup is a huge moral weight. Understand this and make a conscious effort not to be ground down by it, just as you'd be careful to bend at the knees when picking up a heavy box.**

**12. Don't give up.**

**Even if you get demoralized, don't give up. You can get surprisingly far by just not giving up. This isn't true in all fields. There are a lot of people who couldn't become good mathematicians no matter how long they persisted. But startups aren't like that. Sheer effort is usually enough, so long as you keep morphing your idea.**

**13. Deals fall through.**

**One of the most useful skills we learned from Viaweb was not getting our hopes up. We probably had 20 deals of various types fall through. After the first 10 or so we learned to treat deals as background processes that we should ignore till they terminated. It's very dangerous to morale to start to depend on deals closing, not just because they so often don't, but because it makes them less likely to.**

**Having gotten it down to 13 sentences, I asked myself which I'd choose if I could only keep one.**

**Understand your users. That's the key. The essential task in a startup is to create wealth; the dimension of wealth you have most control over is how much you improve users' lives; and the hardest part of that is knowing what to make for them. Once you know what to make, it's mere effort to make it, and most decent hackers are capable of that.**

**Understanding your users is part of half the principles in this list. That's the reason to launch early, to understand your users. Evolving your idea is the embodiment of understanding your users. Understanding your users well will tend to push you toward making something that makes a few people deeply happy. The most important reason for having surprisingly good customer service is that it helps you understand your users. And understanding your users will even ensure your morale, because when everything else is collapsing around you, having just ten users who love you will keep you going.**

**a) Summarize the Paul Graham article in one short paragraph.**

**The most important thing in business is co-founders. How far your business can go half depends on the people you are collaborating with and half depends on how much you understand your users. You should always bring values to your users and make them happy. Besides, insisting is also important, do not give up easily.**

**b) What (if anything) can you apply from this to your life/future/business?**

**I personally love these advices. Because some of my team members say not everyone will use and love our site. However, just as Paul said, even only 10 users love your site, it is still worth it. Another thing I like is the agile plan (business). We should launch our business fast with the minimum feature done, and then keep polishing it according to user’s feedback. That is what I am gonna do for my business.**

# QUESTION D: The Third and Final Paul Graham Essay

**In the Q & A period after a recent talk, someone asked what made startups fail. After standing there gaping for a few seconds I realized this was kind of a trick question. It's equivalent to asking how to make a startup succeed—if you avoid every cause of failure, you succeed—and that's too big a question to answer on the fly.**

**Afterwards I realized it could be helpful to look at the problem from this direction. If you have a list of all the things you shouldn't do, you can turn that into a recipe for succeeding just by negating. And this form of list may be more useful in practice. It's easier to catch yourself doing something you shouldn't than always to remember to do something you should.**

**In a sense there's just one mistake that kills startups: not making something users want. If you make something users want, you'll probably be fine, whatever else you do or don't do. And if you don't make something users want, then you're dead, whatever else you do or don't do. So really this is a list of 18 things that cause startups not to make something users want. Nearly all failure funnels through that.**

**1. Single Founder**

**Have you ever noticed how few successful startups were founded by just one person? Even companies you think of as having one founder, like Oracle, usually turn out to have more. It seems unlikely this is a coincidence.**

**What's wrong with having one founder? To start with, it's a vote of no confidence. It probably means the founder couldn't talk any of his friends into starting the company with him. That's pretty alarming, because his friends are the ones who know him best.**

**But even if the founder's friends were all wrong and the company is a good bet, he's still at a disadvantage. Starting a startup is too hard for one person. Even if you could do all the work yourself, you need colleagues to brainstorm with, to talk you out of stupid decisions, and to cheer you up when things go wrong.**

**The last one might be the most important. The low points in a startup are so low that few could bear them alone. When you have multiple founders, esprit de corps binds them together in a way that seems to violate conservation laws. Each thinks "I can't let my friends down." This is one of the most powerful forces in human nature, and it's missing when there's just one founder.**

**2. Bad Location**

**Startups prosper in some places and not others. Silicon Valley dominates, then Boston, then Seattle, Austin, Denver, and New York. After that there's not much. Even in New York the number of startups per capita is probably a 20th of what it is in Silicon Valley. In towns like Houston and Chicago and Detroit it's too small to measure.**

**Why is the falloff so sharp? Probably for the same reason it is in other industries. What's the sixth largest fashion center in the US? The sixth largest center for oil, or finance, or publishing? Whatever they are they're probably so far from the top that it would be misleading even to call them centers.**

**It's an interesting question why cities become startup hubs, but the reason startups prosper in them is probably the same as it is for any industry: that's where the experts are. Standards are higher; people are more sympathetic to what you're doing; the kind of people you want to hire want to live there; supporting industries are there; the people you run into in chance meetings are in the same business. Who knows exactly how these factors combine to boost startups in Silicon Valley and squish them in Detroit, but it's clear they do from the number of startups per capita in each.**

**3. Marginal Niche**

**Most of the groups that apply to Y Combinator suffer from a common problem: choosing a small, obscure niche in the hope of avoiding competition.**

**If you watch little kids playing sports, you notice that below a certain age they're afraid of the ball. When the ball comes near them their instinct is to avoid it. I didn't make a lot of catches as an eight year old outfielder, because whenever a fly ball came my way, I used to close my eyes and hold my glove up more for protection than in the hope of catching it.**

**Choosing a marginal project is the startup equivalent of my eight year old strategy for dealing with fly balls. If you make anything good, you're going to have competitors, so you may as well face that. You can only avoid competition by avoiding good ideas.**

**I think this shrinking from big problems is mostly unconscious. It's not that people think of grand ideas but decide to pursue smaller ones because they seem safer. Your unconscious won't even let you think of grand ideas. So the solution may be to think about ideas without involving yourself. What would be a great idea for someone else to do as a startup?**

**4. Derivative Idea**

**Many of the applications we get are imitations of some existing company. That's one source of ideas, but not the best. If you look at the origins of successful startups, few were started in imitation of some other startup. Where did they get their ideas? Usually from some specific, unsolved problem the founders identified.**

**Our startup made software for making online stores. When we started it, there wasn't any; the few sites you could order from were hand-made at great expense by web consultants. We knew that if online shopping ever took off, these sites would have to be generated by software, so we wrote some. Pretty straightforward.**

**It seems like the best problems to solve are ones that affect you personally. Apple happened because Steve Wozniak wanted a computer, Google because Larry and Sergey couldn't find stuff online, Hotmail because Sabeer Bhatia and Jack Smith couldn't exchange email at work.**

**So instead of copying the Facebook, with some variation that the Facebook rightly ignored, look for ideas from the other direction. Instead of starting from companies and working back to the problems they solved, look for problems and imagine the company that might solve them. What do people complain about? What do you wish there was?**

**5. Obstinacy**

**In some fields the way to succeed is to have a vision of what you want to achieve, and to hold true to it no matter what setbacks you encounter. Starting startups is not one of them. The stick-to-your-vision approach works for something like winning an Olympic gold medal, where the problem is well-defined. Startups are more like science, where you need to follow the trail wherever it leads.**

**So don't get too attached to your original plan, because it's probably wrong. Most successful startups end up doing something different than they originally intended—often so different that it doesn't even seem like the same company. You have to be prepared to see the better idea when it arrives. And the hardest part of that is often discarding your old idea.**

**But openness to new ideas has to be tuned just right. Switching to a new idea every week will be equally fatal. Is there some kind of external test you can use? One is to ask whether the ideas represent some kind of progression. If in each new idea you're able to re-use most of what you built for the previous ones, then you're probably in a process that converges. Whereas if you keep restarting from scratch, that's a bad sign.**

**Fortunately there's someone you can ask for advice: your users. If you're thinking about turning in some new direction and your users seem excited about it, it's probably a good bet.**

**6. Hiring Bad Programmers**

**I forgot to include this in the early versions of the list, because nearly all the founders I know are programmers. This is not a serious problem for them. They might accidentally hire someone bad, but it's not going to kill the company. In a pinch they can do whatever's required themselves.**

**But when I think about what killed most of the startups in the e-commerce business back in the 90s, it was bad programmers. A lot of those companies were started by business guys who thought the way startups worked was that you had some clever idea and then hired programmers to implement it. That's actually much harder than it sounds—almost impossibly hard in fact—because business guys can't tell which are the good programmers. They don't even get a shot at the best ones, because no one really good wants a job implementing the vision of a business guy.**

**In practice what happens is that the business guys choose people they think are good programmers (it says here on his resume that he's a Microsoft Certified Developer) but who aren't. Then they're mystified to find that their startup lumbers along like a World War II bomber while their competitors scream past like jet fighters. This kind of startup is in the same position as a big company, but without the advantages.**

**So how do you pick good programmers if you're not a programmer? I don't think there's an answer. I was about to say you'd have to find a good programmer to help you hire people. But if you can't recognize good programmers, how would you even do that?**

**7. Choosing the Wrong Platform**

**A related problem (since it tends to be done by bad programmers) is choosing the wrong platform. For example, I think a lot of startups during the Bubble killed themselves by deciding to build server-based applications on Windows. Hotmail was still running on FreeBSD for years after Microsoft bought it, presumably because Windows couldn't handle the load. If Hotmail's founders had chosen to use Windows, they would have been swamped.**

**PayPal only just dodged this bullet. After they merged with X.com, the new CEO wanted to switch to Windows—even after PayPal cofounder Max Levchin showed that their software scaled only 1% as well on Windows as Unix. Fortunately for PayPal they switched CEOs instead.**

**Platform is a vague word. It could mean an operating system, or a programming language, or a "framework" built on top of a programming language. It implies something that both supports and limits, like the foundation of a house.**

**The scary thing about platforms is that there are always some that seem to outsiders to be fine, responsible choices and yet, like Windows in the 90s, will destroy you if you choose them. Java applets were probably the most spectacular example. This was supposed to be the new way of delivering applications. Presumably it killed just about 100% of the startups who believed that.**

**How do you pick the right platforms? The usual way is to hire good programmers and let them choose. But there is a trick you could use if you're not a programmer: visit a top computer science department and see what they use in research projects.**

**8. Slowness in Launching**

**Companies of all sizes have a hard time getting software done. It's intrinsic to the medium; software is always 85% done. It takes an effort of will to push through this and get something released to users.**

**Startups make all kinds of excuses for delaying their launch. Most are equivalent to the ones people use for procrastinating in everyday life. There's something that needs to happen first. Maybe. But if the software were 100% finished and ready to launch at the push of a button, would they still be waiting?**

**One reason to launch quickly is that it forces you to actually finish some quantum of work. Nothing is truly finished till it's released; you can see that from the rush of work that's always involved in releasing anything, no matter how finished you thought it was. The other reason you need to launch is that it's only by bouncing your idea off users that you fully understand it.**

**Several distinct problems manifest themselves as delays in launching: working too slowly; not truly understanding the problem; fear of having to deal with users; fear of being judged; working on too many different things; excessive perfectionism. Fortunately you can combat all of them by the simple expedient of forcing yourself to launch something fairly quickly.**

**9. Launching Too Early**

**Launching too slowly has probably killed a hundred times more startups than launching too fast, but it is possible to launch too fast. The danger here is that you ruin your reputation. You launch something, the early adopters try it out, and if it's no good they may never come back.**

**So what's the minimum you need to launch? We suggest startups think about what they plan to do, identify a core that's both (a) useful on its own and (b) something that can be incrementally expanded into the whole project, and then get that done as soon as possible.**

**This is the same approach I (and many other programmers) use for writing software. Think about the overall goal, then start by writing the smallest subset of it that does anything useful. If it's a subset, you'll have to write it anyway, so in the worst case you won't be wasting your time. But more likely you'll find that implementing a working subset is both good for morale and helps you see more clearly what the rest should do.**

**The early adopters you need to impress are fairly tolerant. They don't expect a newly launched product to do everything; it just has to do something.**

**10. Having No Specific User in Mind**

**You can't build things users like without understanding them. I mentioned earlier that the most successful startups seem to have begun by trying to solve a problem their founders had. Perhaps there's a rule here: perhaps you create wealth in proportion to how well you understand the problem you're solving, and the problems you understand best are your own.**

**That's just a theory. What's not a theory is the converse: if you're trying to solve problems you don't understand, you're hosed.**

**And yet a surprising number of founders seem willing to assume that someone, they're not sure exactly who, will want what they're building. Do the founders want it? No, they're not the target market. Who is? Teenagers. People interested in local events (that one is a perennial tarpit). Or "business" users. What business users? Gas stations? Movie studios? Defense contractors?**

**You can of course build something for users other than yourself. We did. But you should realize you're stepping into dangerous territory. You're flying on instruments, in effect, so you should (a) consciously shift gears, instead of assuming you can rely on your intuitions as you ordinarily would, and (b) look at the instruments.**

**In this case the instruments are the users. When designing for other people you have to be empirical. You can no longer guess what will work; you have to find users and measure their responses. So if you're going to make something for teenagers or "business" users or some other group that doesn't include you, you have to be able to talk some specific ones into using what you're making. If you can't, you're on the wrong track.**

**11. Raising Too Little Money**

**Most successful startups take funding at some point. Like having more than one founder, it seems a good bet statistically. How much should you take, though?**

**Startup funding is measured in time. Every startup that isn't profitable (meaning nearly all of them, initially) has a certain amount of time left before the money runs out and they have to stop. This is sometimes referred to as runway, as in "How much runway do you have left?" It's a good metaphor because it reminds you that when the money runs out you're going to be airborne or dead.**

**Too little money means not enough to get airborne. What airborne means depends on the situation. Usually you have to advance to a visibly higher level: if all you have is an idea, a working prototype; if you have a prototype, launching; if you're launched, significant growth. It depends on investors, because until you're profitable that's who you have to convince.**

**So if you take money from investors, you have to take enough to get to the next step, whatever that is. Fortunately you have some control over both how much you spend and what the next step is. We advise startups to set both low, initially: spend practically nothing, and make your initial goal simply to build a solid prototype. This gives you maximum flexibility.**

**12. Spending Too Much**

**It's hard to distinguish spending too much from raising too little. If you run out of money, you could say either was the cause. The only way to decide which to call it is by comparison with other startups. If you raised five million and ran out of money, you probably spent too much.**

**Burning through too much money is not as common as it used to be. Founders seem to have learned that lesson. Plus it keeps getting cheaper to start a startup. So as of this writing few startups spend too much. None of the ones we've funded have. (And not just because we make small investments; many have gone on to raise further rounds.)**

**The classic way to burn through cash is by hiring a lot of people. This bites you twice: in addition to increasing your costs, it slows you down—so money that's getting consumed faster has to last longer. Most hackers understand why that happens; Fred Brooks explained it in The Mythical Man-Month.**

**We have three general suggestions about hiring: (a) don't do it if you can avoid it, (b) pay people with equity rather than salary, not just to save money, but because you want the kind of people who are committed enough to prefer that, and (c) only hire people who are either going to write code or go out and get users, because those are the only things you need at first.**

**13. Raising Too Much Money**

**It's obvious how too little money could kill you, but is there such a thing as having too much?**

**Yes and no. The problem is not so much the money itself as what comes with it. As one VC who spoke at Y Combinator said, "Once you take several million dollars of my money, the clock is ticking." If VCs fund you, they're not going to let you just put the money in the bank and keep operating as two guys living on ramen. They want that money to go to work. At the very least you'll move into proper office space and hire more people. That will change the atmosphere, and not entirely for the better. Now most of your people will be employees rather than founders. They won't be as committed; they'll need to be told what to do; they'll start to engage in office politics.**

**When you raise a lot of money, your company moves to the suburbs and has kids.**

**Perhaps more dangerously, once you take a lot of money it gets harder to change direction. Suppose your initial plan was to sell something to companies. After taking VC money you hire a sales force to do that. What happens now if you realize you should be making this for consumers instead of businesses? That's a completely different kind of selling. What happens, in practice, is that you don't realize that. The more people you have, the more you stay pointed in the same direction.**

**Another drawback of large investments is the time they take. The time required to raise money grows with the amount. When the amount rises into the millions, investors get very cautious. VCs never quite say yes or no; they just engage you in an apparently endless conversation. Raising VC scale investments is thus a huge time sink—more work, probably, than the startup itself. And you don't want to be spending all your time talking to investors while your competitors are spending theirs building things.**

**We advise founders who go on to seek VC money to take the first reasonable deal they get. If you get an offer from a reputable firm at a reasonable valuation with no unusually onerous terms, just take it and get on with building the company. Who cares if you could get a 30% better deal elsewhere? Economically, startups are an all-or-nothing game. Bargain-hunting among investors is a waste of time.**

**14. Poor Investor Management**

**As a founder, you have to manage your investors. You shouldn't ignore them, because they may have useful insights. But neither should you let them run the company. That's supposed to be your job. If investors had sufficient vision to run the companies they fund, why didn't they start them?**

**Pissing off investors by ignoring them is probably less dangerous than caving in to them. In our startup, we erred on the ignoring side. A lot of our energy got drained away in disputes with investors instead of going into the product. But this was less costly than giving in, which would probably have destroyed the company. If the founders know what they're doing, it's better to have half their attention focused on the product than the full attention of investors who don't.**

**How hard you have to work on managing investors usually depends on how much money you've taken. When you raise VC-scale money, the investors get a great deal of control. If they have a board majority, they're literally your bosses. In the more common case, where founders and investors are equally represented and the deciding vote is cast by neutral outside directors, all the investors have to do is convince the outside directors and they control the company.**

**If things go well, this shouldn't matter. So long as you seem to be advancing rapidly, most investors will leave you alone. But things don't always go smoothly in startups. Investors have made trouble even for the most successful companies. One of the most famous examples is Apple, whose board made a nearly fatal blunder in firing Steve Jobs. Apparently even Google got a lot of grief from their investors early on.**

**15. Sacrificing Users to (Supposed) Profit**

**When I said at the beginning that if you make something users want, you'll be fine, you may have noticed I didn't mention anything about having the right business model. That's not because making money is unimportant. I'm not suggesting that founders start companies with no chance of making money in the hope of unloading them before they tank. The reason we tell founders not to worry about the business model initially is that making something people want is so much harder.**

**I don't know why it's so hard to make something people want. It seems like it should be straightforward. But you can tell it must be hard by how few startups do it.**

**Because making something people want is so much harder than making money from it, you should leave business models for later, just as you'd leave some trivial but messy feature for version 2. In version 1, solve the core problem. And the core problem in a startup is how to create wealth (= how much people want something x the number who want it), not how to convert that wealth into money.**

**The companies that win are the ones that put users first. Google, for example. They made search work, then worried about how to make money from it. And yet some startup founders still think it's irresponsible not to focus on the business model from the beginning. They're often encouraged in this by investors whose experience comes from less malleable industries.**

**It is irresponsible not to think about business models. It's just ten times more irresponsible not to think about the product.**

**16. Not Wanting to Get Your Hands Dirty**

**Nearly all programmers would rather spend their time writing code and have someone else handle the messy business of extracting money from it. And not just the lazy ones. Larry and Sergey apparently felt this way too at first. After developing their new search algorithm, the first thing they tried was to get some other company to buy it.**

**Start a company? Yech. Most hackers would rather just have ideas. But as Larry and Sergey found, there's not much of a market for ideas. No one trusts an idea till you embody it in a product and use that to grow a user base. Then they'll pay big time.**

**Maybe this will change, but I doubt it will change much. There's nothing like users for convincing acquirers. It's not just that the risk is decreased. The acquirers are human, and they have a hard time paying a bunch of young guys millions of dollars just for being clever. When the idea is embodied in a company with a lot of users, they can tell themselves they're buying the users rather than the cleverness, and this is easier for them to swallow.**

**If you're going to attract users, you'll probably have to get up from your computer and go find some. It's unpleasant work, but if you can make yourself do it you have a much greater chance of succeeding. In the first batch of startups we funded, in the summer of 2005, most of the founders spent all their time building their applications. But there was one who was away half the time talking to executives at cell phone companies, trying to arrange deals. Can you imagine anything more painful for a hacker? But it paid off, because this startup seems the most successful of that group by an order of magnitude.**

**If you want to start a startup, you have to face the fact that you can't just hack. At least one hacker will have to spend some of the time doing business stuff.**

**17. Fights Between Founders**

**Fights between founders are surprisingly common. About 20% of the startups we've funded have had a founder leave. It happens so often that we've reversed our attitude to vesting. We still don't require it, but now we advise founders to vest so there will be an orderly way for people to quit.**

**A founder leaving doesn't necessarily kill a startup, though. Plenty of successful startups have had that happen. Fortunately it's usually the least committed founder who leaves. If there are three founders and one who was lukewarm leaves, big deal. If you have two and one leaves, or a guy with critical technical skills leaves, that's more of a problem. But even that is survivable. Blogger got down to one person, and they bounced back.**

**Most of the disputes I've seen between founders could have been avoided if they'd been more careful about who they started a company with. Most disputes are not due to the situation but the people. Which means they're inevitable. And most founders who've been burned by such disputes probably had misgivings, which they suppressed, when they started the company. Don't suppress misgivings. It's much easier to fix problems before the company is started than after. So don't include your housemate in your startup because he'd feel left out otherwise. Don't start a company with someone you dislike because they have some skill you need and you worry you won't find anyone else. The people are the most important ingredient in a startup, so don't compromise there.**

**18. A Half-Hearted Effort**

**The failed startups you hear most about are the spectactular flameouts. Those are actually the elite of failures. The most common type is not the one that makes spectacular mistakes, but the one that doesn't do much of anything—the one we never even hear about, because it was some project a couple guys started on the side while working on their day jobs, but which never got anywhere and was gradually abandoned.**

**Statistically, if you want to avoid failure, it would seem like the most important thing is to quit your day job. Most founders of failed startups don't quit their day jobs, and most founders of successful ones do. If startup failure were a disease, the CDC would be issuing bulletins warning people to avoid day jobs.**

**Does that mean you should quit your day job? Not necessarily. I'm guessing here, but I'd guess that many of these would-be founders may not have the kind of determination it takes to start a company, and that in the back of their minds, they know it. The reason they don't invest more time in their startup is that they know it's a bad investment.**

**I'd also guess there's some band of people who could have succeeded if they'd taken the leap and done it full-time, but didn't. I have no idea how wide this band is, but if the winner/borderline/hopeless progression has the sort of distribution you'd expect, the number of people who could have made it, if they'd quit their day job, is probably an order of magnitude larger than the number who do make it.**

**If that's true, most startups that could succeed fail because the founders don't devote their whole efforts to them. That certainly accords with what I see out in the world. Most startups fail because they don't make something people want, and the reason most don't is that they don't try hard enough.**

**In other words, starting startups is just like everything else. The biggest mistake you can make is not to try hard enough. To the extent there's a secret to success, it's not to be in denial about that.**

**a) Summarize the Paul Graham article in one short paragraph.**

**Paul listed out some common reasons why startups fail. We should always focus on our users, which means try to bring maximum values to our customers. You can always hire the best programmers to work for you but if you business is terrible and your user don’t like it, you will most likely fail anyway.**

**b) What (if anything) can you apply from this to your life/future/business?**

**I think this essay is really useful. We have 3 co-founders right now, including me. One is an ASP.NET contractor who is doing business with BC government right now, and another one is from Sage. They all have their work to do, and I have too much school work to do as well, especially in term 3. So the only time we meet each other and discuss our business is during the weekend. You can image that our business is going really slow. The problem we want to solve is e-commerce goods transportation. Because there are huge volume of trade between Chinese customers and North America sellers, therefore after reading this article, I am wondering should I stop the business till I graduate from BCIT. I can imagine I will be much busier as school goes on.**

# QUESTION E: From Forbes.com

**After his junior year at Brigham Young University, Nick Walter, now 25, landed a great summer internship in the Seattle office of Pariveda Solutions, a Dallas-based tech consulting firm. Though he enjoyed the work and liked his clients and colleagues, he felt stifled. Used to jeans and t-shirts, he didn’t like wearing khakis and polo shirts and most of all, he says, “I hated that I had to be at this office every day for X amount of time doing what they said I had to do.”**

**So instead of heading down the career track he’d always expected of himself—he’d envisioned the security of a steady paycheck and benefits—he decided to go to BYU part-time for the next two years, while hiring himself out as a consultant and developing his own apps for the iPhone including seven how-two apps he wrote with a friend. One of them, called simply Weight Lifting Videos, has helped net $1,200 a month.Then he stumbled on a more lucrative possibility.**

**His inspiration: “I was reading the Tim Ferris book, The Four-Hour Work Week,” he recalls. “Ferris was talking about how the best business you can be in is the education business.” Walter had been out of college from April through May of this year, living on the meager income from his how-to apps and consulting work, when Apple announced it was introducing a new programming language, Swift. A light bulb went off in Walter’s head.**

**He could learn Swift and teach it at the same time, videotaping his progress, and sell the recordings as a course. “For the next four days all I did was dive into this programming language,” he says. “It was wake up, eat, learn, teach, eat, learn, teach, eat, sleep. . . I shut off my phone and went into my cave.” He worked in his pajamas, recording his voice and doing screen captures, making 50 short videos in just four days.**

**Then he uploaded them on Udemy, a four-year-old site that makes it easy to put paywalls around content. Based in San Francisco and running on $48 million in venture funding, Udemy offers 20,000 courses, including everything from yoga to photography to a dance style called Popping and Locking. Prices range from free to $895 for “How to Trade Stock Options: Profiting in Up and Down Markets.” Udemy takes a 50% cut if it attracts people to the courses. If course creators bring in people through their own marketing efforts, they keep 97% (Udemy charges a 3% credit card processing fee).**

**Apple released Swift on June 2. Walter submitted the class on June 5 and for 24 hours, offered it for free. There were 1,600 sign-ups that first day. “That blew me away,” he says. The next day he raised the price to $199 and netted $20,000. Within 30 days he had earned $40,000. “That was more than I’d made in the last year,” he says. It helped that Udemy sent out an email blast to 60,000 people who they thought would be interested in the course and offered it for a discounted $29. Since then the course has generated between $3,000 and $5,000 a month, he says.**

**People who took the Swift course started emailing Walter, asking if he could teach a broader class on how to make an iPhone app. This time he decided to make a marketing video called “How to Make a Freakin’ iPhone App,” and put it on Kickstarter, pre-selling the $199 Udemy course for just $29.**

**Jeff Schwarting, an adjunct professor at Brigham Young who teaches a course on launching products, says Walter hit a sweet spot with a clever campaign. “He happens to be in a space that is really really hot,” says Schwarting. “Typically coding videos are really boring. You just look at a black screen and someone is typing on it while you listen to them talk.” In his Kickstarter video, Walter dances around an empty room with a red brick wall to a pounding electric beat, waving his arms and contorting his body, then promises, “this is a class that’s going to take you from ideas on a napkin to a polished app that you can submit to the Apple app store.” Later he dons a blonde wig and proclaims that Swift has nothing to do with pop star Taylor Swift, singing in a bad falsetto.**

**In one month the Kickstarter video proved a huge success, earning him $66,000. Only 41% of Kickstarter projects reach the goals set by those who launch them and just 14% of those earn more than $20,000, says a Kickstarter spokesman. Udemy senior marketing director Shannon Hughes says it’s not unheard of for course creators to make that kind of money in such a short time, but it’s rare for a novice like Walter.**

**What’s he doing with all that money? He pays $350 a month in rent for a house he shares with four roommates in Provo, UT, and eats most of his meals at Chick-fil-A. He bought a used $9,000 Toyota Corolla because he didn’t own a car. The rest he’s saving (his parents, a nurse and a mortgage broker who work near their home in Wilsonville, OR, paid his way at BYU so he has no student loans).**

**He is planning to put up a Kickstarter video this week about his next Udemy course, on programming for the new Apple smartwatch. “When [the programming tools] come out in November I’ll go into isolation for 48 hours and make videos,” he says.**

**He doesn’t mind that his income is on a see-saw. After banking $66,000 in September he has earned just $3,400 in October. He says he’s fine with that because it more than covers his rent and his modest dining habits. “For the time being I’m very happy getting $3,000 a month and then having these crazy spikes that come in every once in awhile,” he says. He hopes to remain self-employed and support himself with a combination of teaching and app development.**

**Schwarting predicts continued success for Walter. “Ninety-nine percent of my students get jobs in their first year out of college,” he says. “I haven’t seen many who can bring in $100,000 in their first year.” For Walter it took only five months.**

**a) Summarize the Forbes article in one short paragraph.**

**It does not matter if can not earn much at the beginning of your business. As long as you can bring value to your customers and earn the much you can cover your basic spend. Sometimes, it is also a good idea to help other people earn money. In Udemy, you can teach students and earn money.**

1. **What (if anything) can you apply from this to your life/future/business?**

**I think just like Forbes, you can achieve much more when you are doing something you like to do. If you spend lots of time forcing yourself to do something you hate, then it is unlikely that you can success during this kind of industry. The hardest thing is to know yourself, like knowing what you like to do. So, like me, I love business, especially e-commerce, and programming. I will spend lots of my time doing this kind of stuff, because I will always stay excited when I am doing what I like.**

**QUESTION F:**

**Write one PHP script that does the following. Demonstrate the script working by taking a video of it and uploading the video with your submission.**

**Upload the video, the code, and a readme.txt file that tells what percentage of the requirements you have met. Work on your own, not with other students.**

**Your application has an HTML form to allow members to sign up – signup.php - that has a textbox for username and another for password, and a file-upload input too. This is a site for members only. Passwords must be exactly 6 characters long. They CANNOT be dictionary words. Usernames cannot be swear words and cannot be names already stored in the text file. New users must enter an avatar too, a jpg or png file (only) which must be at least 600px wide and at least 600 px tall; rename the avatar file name to the username plus appropriate file extension (.jpg or .png). If these rules are followed, then the signup is completed. Use a text file to store the usernames, passwords, and avatar filename (e.g. “/images/avatars/whatever.jpg”). This completes the sign up process. Tell the user they have successfully signed up. Also show a menu. The menu shows links to pages named membersonly1.php, membersonly2.php, and logout.**

**Another page – signin.php – allows users who have already signed up, to sign IN. If a visitor enters a valid username and password (i.e., the data is in the text file), they are allowed into the site. A session makes sure they are not inactive for more than 10 seconds. Once they sign in, a menu is displayed along with their avatar. The menu shows links to pages named membersonly1.php, membersonly2.php, and logout.**

**membersonly1.php and membersonly2.php ensure the user is logged in and not more than 10 seconds have elapsed since they visited other pages (just like the lab we did earlier in the term). The user’s username and avatar are displayed at the top of the page (“Logged in as …”). Each of these pages has a menu. The menu shows links to pages named membersonly1.php, membersonly2.php, and logout.**

**Logout.php simply logs the user out, then shows a link to signin.php**

**The format of the username/password/avatar file is:**

**users.txt:**

**jason,123456,/images/avatars/jason.png**

**albert,aaabbb,/images/avatars/albert.jpg**

**d’arcy,654321,/images/avatars/darcy.png // no apostrophes in file name allowed**

**Your video capture must demonstrate clearly each of the above requirements.**

**When finished, zip all of your files (readme.txt, membersonly1.php, membersonly2.php, signup.php, signin.php, logout.php, lastname.doc [this Word document with your answers to questions A-E inside it], and lastname.avi) into lastname.zip and upload to SHARE IN under the Share In/COMP/3975/midterm before 11:59pm Sunday November 6, 2016.**