IEEE COMPUTER SOCIETY

www.computer.org

A Guide to Making the Most of Your Computer Science Degree

TABLE OF CONTENTS

INTRODUCTION

Charting a career path in a world riddled with uncertainty is a challenge in the best of times; add to that challenge an industry that is in a near-constant state of growth and flux, and you glimpse the benefits and trials confronting emerging professionals in computer science today.

IEEE COMPUTER SOCIETY

LOOKING AHEAD

A Community of Computing Leaders

The IEEE Computer Society is the world's home for computer science, engineering, and technology. A global leader in providing access to computer science research, analysis, and information, the IEEE Computer Society offers a comprehensive array of unmatched products, services, and opportunities for individuals at all stages of their professional careers. Known as the premier organization that empowers the people who drive technology, the IEEE Computer Society offers international conferences, peer-reviewed publications, a unique digital library, and training programs.

Visit computer.org to launch your career.

Whether you are a researcher, a software engineer, or a graduate student, creating a flexible plan for your near- and long-term success is essential, regardless of the difficulties of achieving a clear vision of the landscape ahead.

Mapping a Changing Field

The U.S. Bureau of Labor Statistics projects a 13% growth in computer occupations through 2030, reflecting growing demands for improved security, technologies, applications, and cloud infrastructure.

Opportunity alone, however, does not necessarily create a satisfying career. Achieving such a career depends on your individual passions and skills, as well as the attributes of various jobs. On the latter front, ZDNet offers some insights.

In its assessment of "The Best Computer Science Jobs in 2022," ZDNet factored in projected growth and demand, and remote work options—along with its three most heavily weighted factors: salary, work-life balance, and job satisfaction.

Topping its resulting list of best jobs was cloud engineer, followed by AI research scientist, data scientist, and mobile app developer, while the highest-paying jobs belong to computer and information research scientists and information systems managers.

Mapping a Changing Field

Demand for new hires in such jobs is only growing, yet as the recent IEEE Computer Society's Guide to Freelancing notes, many of these jobs can be done independently, which can offer freedom and unlimited (though unpredictable) income possibilities.

Also, many of the top jobs—with the highest earning potential—require advanced degrees, a factor projected to grow in importance over the next few years.

According to a 2019 BLS report, by 2026, occupations that typically require a master's degree are projected to grow by 16.7%, while those requiring a doctorate will grow by more than 13%.

Creating a Foundation for Expansion

To take advantage of these opportunities, emerging professionals should lay a foundation for advancement by building their knowledge, their network, and their portfolio. In the following, we discuss these three key areas, guided by the experiences of two seasoned CS professionals: Seema Chopra, Boeing India's Technical Fellow for Artificial Intelligence; and James Ivers, a senior member of the technical staff at Carnegie Mellon University's Software Engineering Institute. They then will offer their thoughts on building your career vision and timing your moves, along with some closing advice.

Seema Chopra / Technical Fellow—Artificial Intelligence, Boeing

Dr. Seema Chopra works at Boeing Research and Technology, India and has spent nearly two decades in the field of Artificial Intelligence (AI) and Machine Learning (ML) churning out multiple patents, trade secrets, and publications. She earned her doctorate degree in control engineering from IIT Roorkee, India.

James Ivers / Initiative Lead of Architecture Design, Analysis, and Automation—Carnegie Mellon Software Engineering Institute

James has a BA in computer science and mathematics from Transylvania University and a master's degree in software engineering from Carnegie Mellon University. He is the co-author of *Documenting Software Architectures* and has worked in the field of software architecture for more than 25 years. He has expertise in architecture and design languages, formal methods, and code analysis. Most recently, he is pursuing the application of search-based software engineering and genetic algorithms to recommend ways to refactor existing software.

Creating a Foundation for Expansion

Building Your Knowledge

Although opportunities for on-the-job learning have been complicated by the pandemic and remote work, they are nevertheless extremely important.

According to an April 2021 Salesforce survey of knowledge workers, 80% said that they retain information better when learning on the job than in formal training sessions, and 73% said they prefer learning that occurs in the course of their daily work.

> Use every opportunity to learn something new that will help you go deeper into your field.

Regardless of the delivery method, optimizing your own on-the-job learning will up your game and your future opportunities. In a 2019 Harvard Business Review article, Josh Bersin and Marc Zao-Sanders outlined excellent strategies for strengthening your on-the-job learning capacity, including creating a "to-learn" list; being an active contributor to workplace learning channels such as Slack or SharePoint; and practicing mindfulness—that is, being "aware and present on the job." The latter, they note, can increase your ability to learn and develop as you focus on asking key questions and listening to responses openly—whether you're in a performance review or trying to understand and code a solution to someone's obscure business problem.

Chopra says that when she was coming up, she never passed up opportunities to gain new

knowledge, and that she was always learning complementary skills, concepts, and technology: "Continue learning and upskill yourself," she advises. "Use every opportunity to learn something new that will help you go deeper into your field."

In addition to standard on-the-job knowledge, you can further your learning by participating in brown-bag sessions, attending code reviews, and using off-hours to learn all that you can about your company's tools and technologies beyond your own area. You can also take online or in-person courses to shore up weak areas, including soft skills such as public speaking and interpersonal communications. Finally, like many organizations, the IEEE Computer Society offers webinars to build your knowledge—whether by targeting a particular area, such as career building, or by accessing deeper insights into technical areas through the Distinguished Lecturer series.

Building Your Network

In any field, connecting with people who share your interests is critical to learning and to finding new opportunities.

Such connections can be even more crucial in emerging fields such as visualization and quantum computing. Making these connections ideally starts in college, where discussions with professors and your classmates can lead to collaborations and even life-long professional relationships. But beyond the

campus and your workplace's walls, there are many other rich opportunities to build your professional network.

Conferences are a great place to start. The IEEE Computer Society's website gives you access to a conference calendar, conference proceedings, and past conference reports. Many of these and other conferences offer volunteer opportunities, which can make it easier to meet professionals with common interests than simply walking into an auditorium of strangers.

For helpful tips on conference attendance, see the University of California, Berkeley's Advising Matters primer on what to do before, during, and after an in-person conference to optimize your benefits. For online sessions, Unito interviewed five experts for their views on how to best learn and connect during virtual events.

You can also meet like-minded professionals and share information through memberships in professional organizations. Most, including the IEEE Computer Society, have local chapters and student memberships. They also offer opportunities to volunteer, which lets you learn and connect with others while helping out with publications,

conferences, standards development, membership, technical communities, and so on.

All of these options, along with early years on the job or in research, offer opportunities to establish formal or informal relationships with mentors. Both Chopra and Ivers noted that mentors were crucial to their own development in various ways.

Although Ivers did not have "official" mentors, he looked up to several of his coworkers early on and tried to learn as much as possible from them. "I lucked into working with good people who were willing to spend time answering my questions (probably far too many of them)."

He recommends asking questions, but always trying to be focused and constructive when you do so. As an example, instead of asking, "How would you do this?," he suggests trying something like: "This is what I'm thinking about as a solution (and why); what am I missing?"

He also recommends volunteering to help with new tasks on the job, which is a great way to stretch your boundaries and work with people who

impress you. “I’ve found that most professionals enjoy a good technical discussion, like to solve problems, and like to learn something new. Be sure to share what you’ve learned, too.”

Chopra said she always had mentors around—many of whom she met through work and took on as role models. “Sometimes [I] had multiple mentors—someone from the technical line, someone who thinks like me, someone from the leadership side (management), someone from a complementary role/field,” she says, adding that, “whenever you have the opportunity to build a connection, you need to ask the right set of questions—20 minutes should be enough to gauge if this person would be a great mentor and for you to leave an impression. It’s all about having the right questions ready to spark their interest and be memorable.”

Chopra suggests asking two questions in particular: Does the person inspire you? Would you like to become like him or her in the future? Also, once you’ve chosen a mentor, Chopra recommends that you be prepared for meeting with them and communicate clearly about your expectations.

I’ve found that most professionals enjoy a good technical discussion, like to solve problems, and like to learn something new. Be sure to share what you’ve learned, too.

"In the first meeting, try to grab your mentor's attention," she says. "Do some homework on your mentor's background, try to see if your mentor shares some of the same passions you have, and bring those points in your introduction." She also suggests that you be frank about why you chose the person as a mentor and that you also discuss practical issues, including how often to meet and what a successful relationship will look like.

Finally, while connecting to and being inspired by a mentor is essential, it is also crucial to examine prospective mentors' technical background, including their published research and areas of specialization to ensure that they have the knowledge, experience, and skills to help you excel and meet your goals.

Building Your Portfolio

At some point, you're likely going to want to make a move, either up within your current organization or out into another one.

Whether that's this year or in five years, taking steps now to document your work, research projects, publications, and extracurricular activities will help ensure that you have a strong portfolio and thus stronger prospects.

> I look for candidates who do more than just write some code; I look for candidates who understand the context of their work and what it means to contribute to the project's overall success.

It is not always possible to go into project details if you work for a private company, but you can track general project goals, skills you learned, skills you contributed, and how successful that contribution was. Those are the sorts of questions Ivers asks candidates. "I look for candidates who do more than just write some code; I look for candidates who understand the context of their work and what it means to contribute to the project's overall success."

You can also build your portfolio by contributing to the numerous open-source projects online today. A 2017 CyberCoders article details how to use your GitHub profile to best show off your skills, projects, and account activity. Other options for building your portfolio include joining hackathons; see Devpost for a searchable hackathon database

that lets you filter by length, interest area, and location (in person or online).

Finally, fellowships can be a benefit whether you actually get one or not, as the application process itself not only teaches you how to think and communicate about your work (effectively and concisely), but it also puts your name and a description of your work in front of key professionals. Fellowship examples include the following:

Spotify Technology Fellowship Program

For programmers who are self-taught or are graduates of bootcamps or community colleges.

Guardian Software Engineering Fellowship

For programmers at the beginning of their careers.

12-week remote MLH Fellowship

For aspiring technologists. Three tracks available: open source, software engineering, and production engineering. Now includes slots for non-student professionals.

I came up before GitHub, but as a hiring manager, it is one of the things that I look for from candidates.

All these activities can make a difference when you are looking for that next-level position. As Ivers notes, "I came up before GitHub, but as a hiring manager, it is one of the things that I look for from candidates."

He added that he always asks a question of job candidates coming out of college that often surprises them: "I ask them to tell me about their experience working with other people's code. Software engineering isn't an individual activity, and most professionals spend a lot of time working in other people's code. Starting from code that someone else wrote (someone who may not be available to answer any questions, no less) is a different experience than writing code from scratch. It's a terribly important skill to learn."

Developing Your Career Vision

Whether, like Ivers, you've known since childhood that computers are your thing or you're simply analytically inclined and want a career in an industry with high pay and projected job growth, once you decide on CS, there are various ways to chart a course forward.

One way to clarify your career vision is to get experience in different environments—from research labs to large companies to small start-ups. It's a path that Ivers himself took, and one that he recommends.

> I encourage those who are still undecided to put their interests to non-trivial tests. Get a job or internship that pushes you and learn from that experience. Pursuing personal projects is good, and encouraged, but not the same as working on a team or putting your work out for others to use and judge.

"There's a difference between what you learn in the classroom (certainly very important) and the experience of professional software development. Gaining that experience early helped me to confirm my path," he says, adding that exploring your options is important, but that you should pursue options with a defined purpose. "I encourage those who are still undecided to put their interests to non-trivial tests. Get a job or internship that pushes you and learn from that experience. Pursuing personal projects is good, and encouraged, but not the same as working on a team or putting your work out for others to use and judge."

Chopra knew by high school that she wanted to be an engineer; by 2001, she was applying fuzzy logic to a clothes washer to see how AI and ML would impact a practical machine. She recommends that emerging professionals and researchers let curiosity lead the way.

"If you're keen on a particular field, find a mentor and build a network so you can bring your crazy ideas to it," she says. "If you want to be innovative, you have to be fearless. Be strong enough to build things and test them. Don't be afraid to ask for resources to make this happen."

In terms of emerging CS opportunities, Chopra is most excited about applications research and fundamental research, especially in AI and ML. Both areas will eventually bring many innovations, she says, but at this point such innovations are still far from practical.

She cites explainable AI as a key area that can build trust in AI decision-making among everyday users in areas such as medical diagnosis. Making AI explainable "demystifies what AI is and shows how it comes to its conclusion," she says, adding that it opens "the black box, helping people understand how things are happening inside and why they're valid."

Ivers is most excited about the basic building blocks available to researchers and

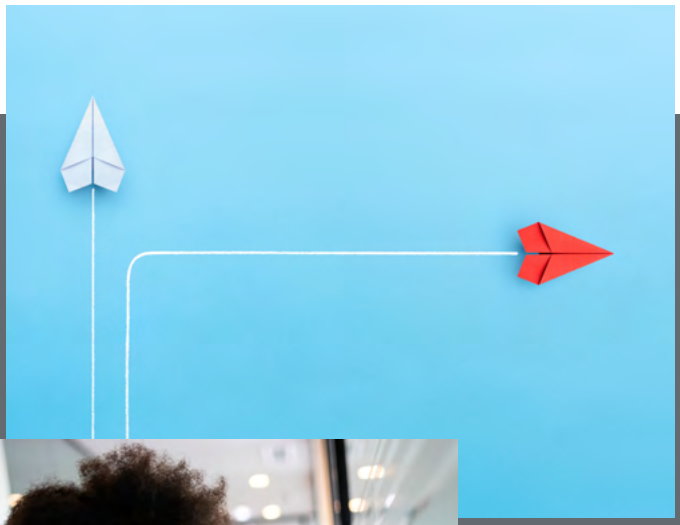
> If you want to be innovative, you have to be fearless. Be strong enough to build things and test them. Don't be afraid to ask for resources to make this happen.

practitioners alike—the "raw tools, frameworks, and libraries that are readily available today provide incredible opportunities to get started on interesting projects quickly."

Although he notes that he is cautious about chasing technology hype, he sees many exciting technologies and ideas on the horizon. Like Chopra, however, he sees a lot of work ahead before they become practical. "Learning to judge the suitability of new technologies for different problems is a skill, like many others."

Key to this judgment, he says, is an understanding of the traits and features of existing technologies, which "can be a big advantage, allowing an individual to quickly learn to use, accommodate, or even work around quirks without having to relearn the lessons of the past."

Timing Your Next Move



When it comes to leaving an organization, timing is everything.

Leave too soon, and you may burn bridges and compromise budding connections; stay too long—particularly if you are unsure it's the right area—and you can miss out on opportunities to find your passion. As Chopra acknowledges, there are pros and cons to both approaches. Still, in her view, establishing yourself within a single company is a better plan for the long haul for several reasons. It allows you to secure more references and log those 10,000 hours required for expertise without the disruptions that job changes bring. Also, she says,

I pursued a path that included large-company commercial work, academic research, a start-up, and an extended stay in applied research and development. I learned something different from each and they have enriched my perspective.

you will make up any short-term loss of income gains over time. Most important, however, are the connections you form and the trust you build among colleagues.

"When you stay with a company, your name, face, work become familiar with your colleagues and employers," she says. "Staying with a company, you see the good/bad, up and down times, and it can be rewarding in the long term."

Ivers, who himself explored various areas before settling in, says there is probably not one right path. "Context matters," he says. "I would generally encourage sticking around a project long enough to see the results of your work." This in itself is an opportunity for learning, including how users received it, what you did right and wrong, and what had to change for subsequent releases. "As a hiring manager, frequent job changes on a resume can be a red flag," says Ivers. "That said, I pursued a path that included large-company commercial work, academic research, a start-up, and an extended stay in applied research and development. All have their charms and were rewarding in different ways. I learned something different from each, and seeing the differences has enriched my perspective."

Final Thoughts and Advice

If anything comes up repeatedly, whether considering the foundational elements of an early CS career or discussing the topic with seasoned professionals, it is the need to always be learning and connecting with others. This can happen in seminars, at conferences, through open-source coding adventures, or, of course, on the job.

Chopra says that when she was starting out, she was always thinking about what more she could do and learn. "Whenever you have an internship or free time, use that time to learn something new, especially when the materials are there," she says, adding that it is also important to participate in culture and other activities, such as professional societies, that allow you to develop leadership and soft skills. "Use all these times to connect and build relationships with people."

As for final advice, Ivers suggests: "Enjoy the ride? It's a wonderful field in which we are always learning and often able to quickly see the results of our work. Balance everything you learn in the classroom with practical

application on a job. Get internships while still in school so that you can learn both aspects concurrently. Don't assume that the newest idea you just read about is really a new idea; there's a lot to learn from earlier incarnations of a technology."

Chopra advises that, on the technical front—and regardless of industry—you always focus on developing and using three essential tools:

1. **Logic/Reasoning**
   (critical thinking)
2. **Analytics and Problem Solving**
   (solving issues empirically)
3. **Creativity**
   (innovative thinking)

"Take calculated risks all the time," Chopra says. "If you're crazy to do something, ask for the resources to get it done. Get out of your comfort zone, think about how you can do things differently."

Resources

Opportunities for Continued Learning

1 Articles

- Computer and Information Technology Occupations (U.S. Bureau of Labor Statistics)
- The Best Computer Science Careers in 2022 (ZD Net)
- Occupational Employment Projections Through the Perspective of Education and Training (U.S. Bureau of Statistics)
- Survey - 59% of US Workforce Reports Fewer Workplace Learning Opportunities Since Pandemic Began (Salesforce News)
- Making Learning a Part of Everyday Work (HBR)
- Getting the Most Out of a Professional Conference (UC Berkeley)
- Attending a Virtual Conference for the First Time? 5 Experts Share Their Tips (Unito)
- How to Effectively Use GitHub as a Job Seeker (CyberCoders)

2 Hackathons & Fellowships

- The Home for Hackathons (Devpost)
- Technology Fellowships (Spotify)
- Guardian Software Engineering Fellowship (The Guardian)
- MLH Fellowship for Aspiring Technologists (MLH Fellowship)

3 Computer Society Professional Resources

- The Computer Science Professional's Guide to Freelancing
- Build Your Career Webinar Series
- Discover Computer Science Conferences to Attend
- Connect with Computer Scientists in Your Field - Join a Professional Chapter
- Opportunities to Get Involved - Lend Your Experience to Drive the Future of Your Field
- International Network of Professionals Creating the Latest Advances in Software, Security, AI, Hardware, Systems, and HPC

IEEE COMPUTER SOCIETY

www.computer.org