Software Industry Readiness for Fresh Graduates

Ferdous Mahmud Shaon
Chief Operating Officer
Cefalo Bangladesh Ltd.

IT Recruitment

- I’ve recruited more than 100 software engineers
  - Freshers to 15 years experienced developers
  - Last year: 30 software engineers
  - Interviewed more than 500 candidates

- Biggest Challenge in IT Recruitment:
  - Lack of enough skilled software engineers
  - Demand > Supply
  - Effort > Output

IT Recruitment Challenge

- Whenever we publish a job ad:
  - We get hundreds of CVs
  - Shortlisted: 20 candidates
  - Initial screening + phone interview: max 10 candidates
  - Final interview: if lucky, we can select maximum 1 to 2 candidates, sometimes we are unlucky

IT Graduates

- Number of public universities: 40+
- Number of private diversities: 100+

- More than 1 lac IT graduates passed per year

- But more than 30% are unemployed
- More than 50% don’t get suitable jobs

Employment Challenge for IT Grads

- What is the primary reason of un-employment of IT graduates?

- Why some of us are not getting suitable jobs even after completing IT graduation?

LACK OF NECESSARY SKILLS

Primary Skill of Software Engineer

What is the **most important skill** of an IT graduate for Software Engineering job?

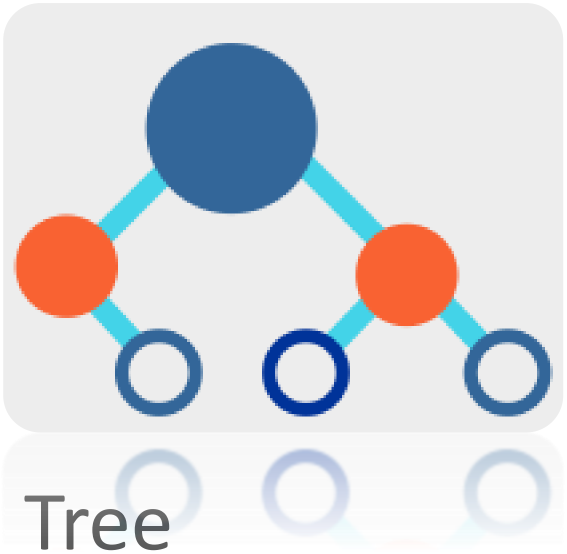
GOOD PROGRAMMING SKILL

Programming Skill

- Programming is a **passion**, not only a profession
- Must have eagerness to **solve complex problems**
- Must have thorough understanding of **algorithm** and **data structure**

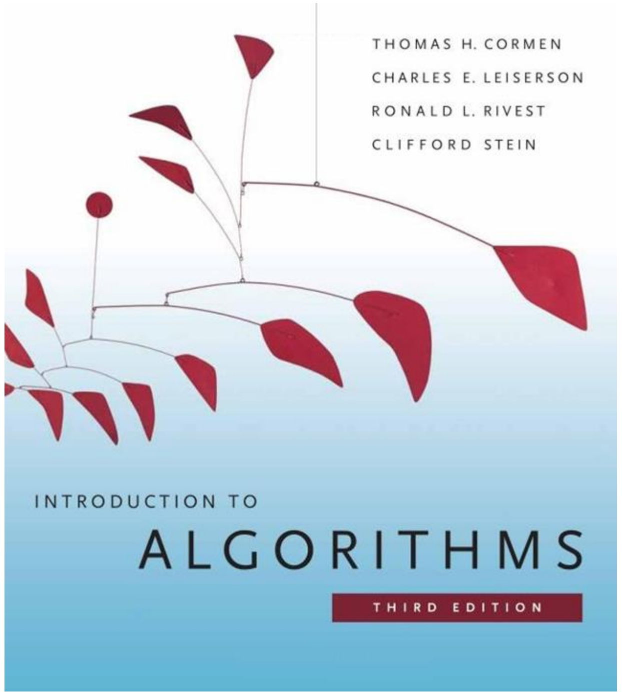
Learn Data Structures

- Array, List, ArrayList, LinkedList
- Set, HashSet, TreeSet
- Map, HashMap, HashTable
- Stack, Queue
- Tree, Binary Tree, Binary Search Tree
- Graphs, Adjacency Matrix, Adjacency List



Learn Algorithms

- Insertion, Deletion, Traversal, Searching and Sorting within Collection
- Sorting: Bubble, Insertion, Merge
- Searching: Linear Search, Binary Search, BFS, DFS
- Recursion
- String manipulation
- Time and space Complexity, Big-O notation



Train your Brain

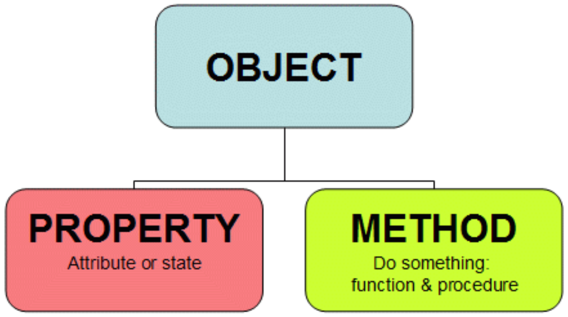
- Learn algorithms as many as you can. Implement a few.
- Can you run a multiple recursion in your brain?
- Sit and think before writing code
- Read complex stuff, make an habit to digest things that are hard to swallow.

Solve Programming Problems

- ACM Problems
- uva.onlinejudge.org
- Codeforces - codeforces.com
- Top Coder - topcoder.com
- Hacker Rank - hackerrank.com

Learn OOP & Design Patterns

- Industry runs on OOP
- Learn to think in Objects, not in Methods
- Learn at least **one OOP language** well



- **SOLID Principal**, get a strong hold on it. SOLID is Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion.

- Read GoF's "Design Patterns: Elements of Reusable Object-Oriented Software". Build a pattern **vocabulary**.

Source Control System

- Never, ever work without a Source Control.
- Not even when you are working alone.
  - Solution of Online Problem Sites
  - All university assignments and projects
  - Personal pet projects with commit history
- Use a distributed version control such as Git (GitHub / BitBucket / GitLab)
- Have good understanding of Git (i.e. push, pull, merge)

Showcasable Project

- Must have one large demoable project
  - could be web or mobile app or both
- Tentative project length: at least 01 year
- Preferably solve some real world problem instead of traditional university projects
- Recommended to use **OOP approach**
- Recommended architecture: separate application for back-end (REST API) and front-end (web / mobile)
- Codes must be in **source control** with commit history

Nice to Have

- Contribute to StackOverflow, CodeProject etc.
  - Reputation points for asking technical questions & giving answers
  - Good for knowledge sharing, plus point to attract employers
- Write Technical Articles, Blog, Online Portfolio
  - Can write articles in Medium, CodeProject etc.
  - Can also build personal portfolio site on personal domain or github.io
- Contribute to Open Source Projects
  - Mozilla Firefox?
  - Java: Apache Commons, Spring, Guava
  - Microsoft: Visual Studio Code Editor
  - Start with small things, like documentation, fixing bugs

LinkedIn Profile

- Must be updated, with proper profile picture
- Use suitable Title
  - Full Stack Software Engineer | Competitive Programmer | Interested in Java based application development
  - Software Engineer | Experienced in C#, ASP.NET Web API, REST, EFF
- List of all projects - university projects, personal projects
  - Summary, Tools & Technologies, Repo, Project Link & Demo
- All the relevant presentations
- Achievements, Certifications, Awards
- Recommendation from friends, teachers, advisers

Online Presence

- Online Problem Solving
- Git repository
- StackOverFlow, CodeProject
- Technical blog, Online Portfolio
- Contribution to Open Source Projects
- LinkedIn Profile

Job Application

- Read the **Job description & requirements** thoroughly before applying for any job

- **Do not apply**, if you do not fulfill the most of the job requirements

- Do not use any **informal email address** or name (i.e. cooldude@gmail.com)

CEFALO

Job Application (cont.)

- Email body should not be blank
  - Justify in email why you are a good fit for this open position
- Attach your CV and other documents, as per job requirements
- Attached CV File should be
  - **PDF format**
  - **Named in a formal manner** and must comply job application requirement

Curriculum Vitae (CV)

- Most important for getting an interview call
  - First impression to the Employer
- Ideally 1 page, not more than 2 pages.
  - might be compact version of your LinkedIn Profile
- We generally don't spend more than 15 seconds for initial screening
- Use standard fonts (Arial / Times New Roman / Calibri)

MUST have in your CV…

- Online Presence (LinkedIn, GitHub, SOFlow, Tech Blog)
  - only relevant links, that adds value
- Professional Experience (if any)
  - recent one comes first.
  - Include primary responsibilities, technology used, links
- Technical skills
  - sorted based on your knowledge & relevance
- Professional / Personal Projects
  - Only recent & relevant projects, you know very well and you were actively involved
- Education (may skip SSC & HSC degrees)
- Relevant Certifications & Awards

DON’Ts in a CV…

- Quantifying your skill level (i.e. Expert in Java or C#)
- Every single buzzword you’ve heard of. (i.e. “big-data”, “AI”, “machine learning”)
- Any project that you don't know well
- Irrelevant skills (i.e MS-Word, MS-Excel etc.)
- Irrelevant social media links (i.e. FB or Twitter)
- More than three fonts and three font sizes (for readability)
- Any other format than PDF
  - contents must be machine readable, not in image
- Irrelevant personal infatuation
  - religion, marital status, parent's name, DOB, blood group etc.

Programmers VS Developers

- All of us can code.
- Some of us are brilliant programmers.
- Some of us are GEEKS.
- Some of us have solved hundreds of ACM problems
- But NONE of us are developers here!

PROGRAMMERS CAN ONLY CODE

DEVELOPERS CAN DELIVER

Delivery Matters

- How many of us did a project that we did not finish?
- How many of us did a project we never deployed?
- How many of us did a project no one ever used?

You'll never learn to deliver until you join the industry.

BUT

You may not learn to deliver even after joining!

Software is Delivered when it…

- Works under **worst case scenario**
- Solves some actual **business problem**
- Can handle **scalability**
- Provides user a comfortable **User eXperience** (UX)
- Has proper and detailed **documentation**
- Can be **iteratively improved**
- Can be **changed** with less cost
- Can be **modified and maintained** by a person who did not originally developed it

Software Delivery requires…

- OOP and Design Patterns
- Readable & Maintainable Source Code
- Refactoring and Code Smell
- Source Control
- Unit Testing
- Continuous Build & Integration
- Comfortable User eXperience (UX)
- Incremental Software Development (Agile, Scrum, Kanban)

:CEFALO

Learn Daily and Adopt Early

- You are the ONLY one, responsible for your own career, knowledge and personal growth
- Keep learning: new tools & technology, new standards and best practices
- At least follow 5 relevant technical blogs, study technical articles
- Know all the famous people in your platform, follow them, read them.
- Adopt and learn any new relevant technology as soon as it is released.

Thank You!

Question & Answers

?