Why Problem Solving?

sandeep@beingzero.in

Industry & Freshers

- Hiring fresher is the most cost-effective way for an IT Company to increase their workforce.
- But lots of companies consider it to be risky, because there are **challenges** involved, in hiring freshers.

Challenge 1 – Practical Knowledge

Lack of

practical knowledge

of

languages/technologies;

Challenge 2 - Industry Tools

Lack of

exposure

to industry tools/practices/culture.

Being Zero

Challenge 3 – Client Expectations

Lack of
confidence
to handle ambiguous
situations
with customers.

Challenge 4 – Training

Require companies to

spend

(time & money)

to train them

Problem with Education

- More theoretical in nature.
- Typically,
  - Focus on fundamentals is less.
  - Or Linkage of fundamentals to real world isn't emphasized.

Desired Skills

- Skill Stages
  - **Problem Solving Skill**
    - Programming Language
    - Data Structures
    - Ability to convert ideas to Code
  - Base for Advanced Skills
    - **Object Oriented Programming**
    - Software Architecture
    - User Interface design
  - Professional Skills
    - Integration of technologies (Database, Cloud, Mobile, ASP.NET, JSP etc.)
    - **Presentation & Etiquettes**
    - Analysis

Software World

- Programmer
  - Given a well defined Programming problem
    - Input, Output.
    - Writes Code that solves it.
  - Skills Required: Problem Solving Skills (Language, Data Structures, Algorithms)
- Developer
  - Given a real-world problem, designs an object oriented solution and produces working code.
  - Collaborates with other developers & testers to use:
    - Uses Source Code Control Software
    - Design classes and interfaces
    - Bug fixing software
- Architect
  - Interacts with Customers to analyze the problem.
  - Comes up with overall system design which is
    - Extensible
    - Robust
    - Scalable
    - Easy to Maintain
  - Deals with multiple technologies (Databases, Cloud, Mobile, Design Patterns)

What are we going to cover?

- Learn
    - Master Language Fundamentals.
    - Master Data Structures
    - Learn to come up with logic.
    - Transform logic into working code.
- Build
    - Pick programming problem from real world software.
    - Discuss ways to solve it.
    - Write code that solves the problem.
- Grow
    - Assignment and Discussion of interview programming problems.
    - Presentation and Interviewing skills.
    - One end to end solution to a gaming problem.