



PONDICHERRY UNIVERSITY – PUDUCHERRY **PLACEMENT CELL & WIPRO TECHNOLOGIES**

Jointly Organizes

**“Project Campus 2012”
Training Programme for
Wipro Recruits
& Presentation of
Wipro Offer Letters**

**27/Feb/12 to 02/Mar/12
(5 Days)**

**@ Sem Hall of
Dept. of CS and ES,
S&H Block II, PU**

**For all 140 Wipro
Recruited Students**

**Inauguration:
JN Auditorium
08.30 AM, 27/Feb/12**

Project Campus 2012

About project Campus

Transitioning from a Campus life to a corporate environment is a key milestone in the transformation process of any individual. A student faces several challenges while moving from an individual to a team-based mode compounded with the pressure of day-to-day performance. This calls for a structured learning process, where they are equipped with the skills and competencies required to succeed in the corporate environment.

A study conducted among the managers of new campus recruits at Wipro, highlighted the need to nurture certain key skills among the recruits.

Project Campus, also referred as Jumpstart Program, has been designed to bridge this gap. Jumpstart is a five-day program with the objective of augmenting Wipro recruits with problem-solving and programming skills to become 'work- ready IT professionals'

Problem-solving skills will help them take decisions in the real world by using a scientific approach to arrive at an optimum solution by narrowing down a range of possible solutions and keeping the customer requirements in mind. Programming skills, will equip them to find effective solutions using an algorithmic approach. Thus, recruits learn to be customer-orientated and scientific in their approach to solving a problem and in turn offer effective and efficient solutions.

Jumpstart will cover the following topics:

1. Problem-Solving Skills

- Problem Domains
- Problem Definition & Representation
- Problem Analysis & Solution Techniques
- Choice of Techniques & Solution Verification
- Lateral and Creative Thinking approaches

2. Programming Skills

- Programming Essentials
- Patterns in Programming Problems
- Program Construction and Testing
- Data Structure and Algorithm Techniques
- Programming Principles, Methodologies and Practices

Training Methodology

The training methodology in Jumpstart includes a variety of learning tools to optimize the learning process to make it dynamic and vibrant. Some of these are:

- Lectures / Presentations / Discussions
- Video based discussions
- Case Studies / Examples / Exercises
- Role Plays
- Assessments

Through a dynamic and exercise-based learning environment and a diverse combination of learning methodologies, Jumpstart helps a new recruit to move up the learning curve quickly and face the new frontier of professional life in Wipro.

Note

Wipro Faculty will be deployed to conduct the Jumpstart courses at the Academia for 5 days. Courseware for students will be provided by Wipro. Administering of assessment will also be facilitated by Wipro. The role of the Academia here will be to

- Provide required infrastructure that can accommodate a minimum of 40 students
- Audio-Video facilities, OHP & White boards/Black boards
- Identify a student SPOC to help in coordination and logistics enablement during the course

Assessment post Jumpstart

An assessment will be conducted on the 5th day at the end of the Program. It would be an objective type assessment of 1 hour duration.

1st day of the course starts with introduction, course objectives and expectation setting. After that it sets up a discussion around Problem Solving as a skill, shows a simple example where multiple techniques can be used to solve one problem. Next it talks about Problem domains and the types of problems faced by a software engineer after joining the IT industry. A test is conducted through which participants can understand various problems domains and their own approaches to problem solving. Subsequently, the techniques to define a problem and represent a problem are explained with examples.

Course Objectives:

- Introduction, Course Objective and Expectation Settings
- Identify different Problem Domains with examples
- Identify and use Problem Definition Techniques
- Identify and use Problem Representation Techniques

Module-wise Break-Up

Module	Module Topics	Duration
1. Problem Domains	a. Problem Solving as a Skill b. A Simple Problem ü 3 Solution Approaches c. Problem Domains d. A Day in the Life of a Software Engineer	1.5 hours
2. Test	a. Solve 20 objective type Problems b. Debrief on Solutions to know ü Approaches followed by different members ü Demonstrating a right approach ü Understanding the problems are mapped to different domains	1.5 Hours
3. Problem Definition	a. Constituents of a given Problem b. Importance of correct Problem understanding c. Problem Definition Techniques ü Problem Definition Checklist ü MECE Technique d. Example usage & Exercise	1.5 Hours
4. Problem Representation	a. Problem Representation Techniques b. Issue Tree ü Example usage & Exercise c. Use Case Diagrams ü Example usage & Exercise	2 Hours

Reference Materials:

1. Gardner, Martin. *My Best Mathematical and Logic Puzzles* Dover Publications, 1994.
2. Mahajan, Sanjoy. *Street-fighting Mathematics – The art of educated guessing and opportunistic problem solving* MIT Press, 2010.

2nd Day starts with a short quiz and recapitulation of Day 1. Then it takes the Problem Solving process to the next step of Problem Analysis. Thereafter some important Problem Solving Techniques are explained with examples. At the end of the day, the choice of techniques in Problem Solving is explained with a number of exercises where the different techniques learnt can be applied by participants. The participants are then asked to work on a Case study problem as a home work.

Course Objectives:

- Use Problem Analysis technique on a given problem
- Identify different techniques for Problem Solving
- Explain pros-and-cons of different techniques
- Use the above techniques to solve given problems

Module-wise Break-Up

Module	Module Topics	Duration
5. Problem Analysis	a. Problem Analysis Step ü Drill Down Technique b. Exercise to apply the given approach	1.5 hours
6. Problem Solving techniques - part 1	a. Problem Solving Techniques (part 1) ü Symmetry ü Pigeon-hole Principle ü Inductive Reasoning ü Deductive Reasoning ü Recursion b. Example usage for each technique and exercises	1.5 hours
7. Problem Solving techniques - part 2	a. Problem Solving Techniques (part 2) ü 5 Whys ü Deming Cycle b. Example usage for each technique and exercises	1.5 hours
8. Choice of Techniques	a. Pros-and-cons of different techniques b. Solution Trade-offs c. Problem Solving Exercises	2 hours

Reference Materials:

1. Gardner, Martin. *My Best Mathematical and Logic Puzzles* Dover Publications, 1994.
2. Smith, Douglas J. *Hdl Chip Design: A Practical Guide for Designing, Synthesizing & Simulating Asics & Fpgas Using Vhdl or Verilog*. Doone Pubns, 1998.

3rd Day starts with a short quiz and recapitulation of Day 2. Then it takes the Problem Solving process to the next step of Solution verification – how to verify and justify a solution to a given problem. Thereafter the course introduces an out-of-the box technique, namely, Lateral-thinking approach and explains the role of such a technique in Problem Solving. The participants present the Case Study solutions from each group. After that few creative approaches to problem solving are explained. The day ends with a discussion on the traits of a good Problem Solver.

Course Objectives:

- § Use Solution Verification techniques for a given problem
- § Use Out-of the Box and creative approaches to Problem Solving
- § Present your Solution Approach to a given Case Study problem
- § Discuss on the Traits of an Effective Problem Solver

Module-wise Break-Up

Module	Module Topics	Duration
9. Solution Verification	a. Solution Verification Techniques <ul style="list-style-type: none"> ü Contradiction ü Induction ü Deduction b. Role of Testing in Software Programming c. Exercises to apply the given techniques	1.5 hours
10. Beyond Logical Thinking	a. Lateral Thinking Approach b. Logical vs Lateral Thinking c. When to use such techniques d. Example usages & Exercise	1.5 hours
11. Case Study Presentation	a. Case Study Presentation <ul style="list-style-type: none"> ü Discussion on solution approaches 	2 hours
12. Creative Approaches to Problem Solving	a. Pair Problem Solving b. Creative Approaches <ul style="list-style-type: none"> ü Brainstorming (Role Play) ü SCAMPER ü Disney Creative Strategy ü Kano Model Analysis c. Traits of a good Problem Solver	1.5 hours

Reference Materials:

1. Gardner, Martin. *My Best Mathematical and Logic Puzzles* Dover Publications, 1994.

4th day of the program/workshop provides fundamental concepts of programming by discussing programming process/life cycle, rules and techniques. It also discusses programming problem patterns using examples and provides tips to construct a simple program using pseudo code.

Course Objectives:

- Distinguish Problem Solving and Programming
- Correlate problem solving and programming process
- Describe patterns in programming problems
- Construct Pseudo code and test it

Module-wise Break-Up

Module	Module Topics	Duration
13. Programming Essentials	a. Problem Solving Process b. Problem Solving & Programming ü Examples & Exercises c. Programming Life Cycle d. Programming Rules e. Programming tools ü Algorithm ü Flow charts	1.5 Hours
14. Programming Problems/patterns	a. Programming Problems & Process b. Problem Patterns ü Examples & Exercises c. Programming features d. Program Design - Tips	1.5 Hours
15. Program Construction	a. Problem Solving Tools - Pseudo Code b. Features, Operators, Conventions, I/O operations, Constructs ü Examples & Exercises	2 Hours
16. Program Testing	a. Testing a Pseudo Code b. Dry Run - Trace Table ü Examples & Exercises - Case Studies	1.5 Hours

Reference Materials:

1. Lipschutz, Seymour. *Schaum's Outline Series: Theory and Problems of Data Structures*. New Delhi: McGraw Hill, 1986.

5th day of the program/work shop provides brief overview on programming techniques such as data structures, algorithm techniques, methodologies and best practices. Also, participants are expected to work on given case study to solve the problem and construct a pseudo code before presenting/sharing with the team.

Course Objectives:

- Explain features of various data structures
- Differentiate different algorithm design techniques
- Distinguish programming methodologies
- Highlight programming principles
- Describe best practices followed in programming process

Module-wise Break-Up

Module	Module Topics	Duration
17. Data Structures & Algorithm Techniques	a. Computer Programming <ul style="list-style-type: none"> ü Data types ü Data Structures ü Containers b. Algorithm techniques <ul style="list-style-type: none"> ü Examples & Exercises 	2 Hours
18. Programming Exercise	a. Group Activity <ul style="list-style-type: none"> ü Traffic Light problem 	2 Hours
19. Program Presentation	a. Group Activity <ul style="list-style-type: none"> ü Traffic Light problem 	1 Hour
20. Programming Practices	a. Programming Principles b. Programming Methodologies c. Programming practice	1.5 Hour

Reference Materials:

1. Lipschutz, Seymour. *Schaum's Outline Series: Theory and Problems of Data Structures*. New Delhi: McGraw Hill, 1986.