Problem Solving Techniques

IGATE

Speed.Agility.Imagination

Course Goals and Non Goals

- **Course Goals**
  - To learn about how to write good program by understanding concepts like
    - Readability
    - Maintainability
    - Modularity
    - Defensive programming
    - Algorithm analysis and design
  - To learn about how to write pseudocode in design phase
  - To develop robust programs by performing Code
- **Course Non Goals**
  - To learn any specific language features in this course.

Intended Audience

- **Developers**

Day Wise Schedule

- **Day 1**
  - Lesson 1: Introduction to Program development with Algorithm and pseudocode
  - Lesson 2: Algorithm Analysis and Design
  - **Lesson 3: Algorithm Design Techniques**
- **Day 2**

  Lesson4: Some more data structure

- **Day 3**
  - **More examples from Question bank**
  - **Module end test**

Table of Contents

- **Lesson 1: Introduction to program development with pseudocode**
  - 1.1 Introduction to Programs
  - 1.2 Analyze the problem statement
  - 1.3 Design Phase
  - 1.4 Implementation
  - 1.4 Introduction to Pseudocode
  - 1 .5 Implementation
  - 1.6 Introduction to Pseudocode How to write Pseudocode?
  - 1.7 Usage of variables and operators
  - 1.8 Control structures Conditional Statement
  - 1.9 Arrays
  - 1.10 Introduction to Linked List

Table of Contents

- **Lesson 2: Algorithm Analysis and Design**
  - 2.1 Algorithm Analysis and efficiency
  - 2.2 Measuring Unit for Algorithm
  - 2.3 Order of Growth
    - Asymptotic notations
  - 2.4 Best/Worst/Average case
  - 2.5 Efficiency of algorithm
  - 2.6 Space Efficiency

Table of Contents

- **Lesson 3: Algorithm Design Techniques**
  - 3.1 Algorithm Design Technique
  - 3.2Brute Force
  - 3.3Divide and Conquer
  - 3.4 Decrease and Conquer
  - 3.5 Space and Time Tradeoffs
- **Lesson 4: Other data structures and techniques**
  - 4.1 Stacks
  - 4.2 Queues
  - 4.3 Hashing
  - 4.4 Recursion
  - 4.5 Collection classes in JAVA

Next Step Courses

- **Any programming language**