19CSE100 Problem Solving and Algorithmic Thinking

Course Administrivia and Introduction

Problem Solving and Algorithmic Thinking

L –T - P

2– 1 – 3 – 4 Credits

Course Objectives

- To provide the foundations of algorithmic thinking and problem solving
- To focus on *principles* and *methods* rather than on *systems* and *tools* thus providing transferable skills to any other domain
- To provide foundations for developing computational perspective of one's own discipline

Syllabus

Unit I

- **Problem Solving and Algorithmic Thinking Overview** – problem definition, logical reasoning; Algorithm – What is an algorithm, practical examples, properties, representation, algorithms vs programs.

Unit II

- **Algorithmic thinking** – Constituents of algorithms – Sequence, Selection and Repetition, input-output; Computation – expressions, logic; algorithms vs programs,
- **Problem Understanding and Analysis** – problem definition, input-output, variables, name binding, data organization: lists, arrays etc. algorithms to programs.

Unit III

- **Problem solving with algorithms** – Searching and Sorting, Evaluating algorithms, modularization, recursion.
- C for problem solving – Introduction, structure of C programs, data types, data input, output statements, control structures.

Textbook and References

- Paolo Ferragina and Fabrizio Luccio, "*Computational Thinking: First Algorithms, Then Code*", Springer, 2018.

- Karl Beecher, "*Computational Thinking: A beginner's guide to problem-solving and programming*", BCS, The Chartered Institute for IT, 2017.

- Paul Curzon and Peter William Mcowan, "The *Power of Computational Thinking: Games, Magic And Puzzles To Help You Become A Computational Thinker*", WSPC (EUROPE), 2017.

- David Riley and Kenny A. Hunt, "*Computational Thinking for the Modern Problem Solver*", Chapman and Hall/CRC, 2014.

Course Outcomes(CO)

- CO1: Apply algorithmic thinking to understand, define and solve problems
- CO2: Design and implement algorithm(s) for a given problem
  CO3: Apply the basic programming constructs for problem solving
- CO4: Understand an algorithm by tracing its computational states, identifying bugs and correcting them

Tools Used : Flowgramming

- Official website for flowgramming:
  - https://flowgrammers-org.github.io/flowgramming/

- Flowgramming Manual:
  - https://flowgrammers-org.github.io/flowgramming-manual/

Lab based course

Evaluations

Continuous Assessments-65 marks

End Semester Examination-35 marks

Assessment Pattern

Internal: 65 Marks

1. **Theory: 15 Marks**
    1. Quizzes – 10 Marks (5 Quizzes of 2 Marks each)
    2. Debug / Tutorial – 5 Marks (1 test for 5 marks)
2. **Lab: 30 Marks**
    1. Project – 20 Marks
        1. *Case Study – 10 Marks*
        2. *Implementation – 10 Marks*
    2. Lab Evaluation – 10 Marks (2 labs of 5 marks each)
3. **Mid Term Exam 20 Marks**
    1. Online Test – 10 Marks
    2. Viva – 10 Marks
4. **Strictly one missed Quiz, Lab Evaluation, Missed Periodical**

End Semester - 35 marks

- **Online Exam – 15 Marks**
- **Viva – 20 Marks**

Video

An example for problem solving

Car allergic to Vanilla Ice cream

- Strange complaint @ Pontiac Division of General Motors
- My car won't start after I buy a Vanilla Ice cream
- Correlation does not mean causation

Credits: https://wallhere.com/en/wallpaper/748460

Solution

- Refer PDF