

# Implementation of OBE Approach for Computer Algorithms Course and it's Outcome

Pragati Sawant

Department of Information Technology  
Rajarambapu Institute of Technology  
Sangli, India  
pragati.sawant@ritindia.edu

**Abstract**— Our IT program contains a course, Computer Algorithms (CA) for Second Year of Engineering. This course is newly introduced in the IT curriculum. Before teaching this subject I had interaction with the students who have already studied this subject in the last year. Based on the discussion, I realized that understanding of algorithms is one of the most challenging aspects for the students. Many complicated concepts and algorithms make this course to be a difficult one. Then, I have designed the concept map and OBE approach which determines, what students are expected to achieve after the completion of this course. Also, the activity plan is prepared to implement the OBE approach. The main goal of this paper is to provide an environment that engages the students allowing the construction of knowledge in a meaningful way by incorporating various interactive activities such as poster design, group presentations, animations, Quizzes, etc. This paper presents how the OBE approach for Computer algorithm course is implemented at our Institute.

**Keywords**—computer algorithm; concept map; OBE; outcome; mapping.

## I. INTRODUCTION

Computer Algorithm plays very important role in IT. It is a core foundation stone in IT curriculum. IT specialists are of vital importance as they are acquainted with the technology required to solve the problem. Companies that look to increase their revenues via technology need IT professionals. Therefore, understanding of an algorithm is an important part of being able to effectively apply it to a problem.

Almost everything that we do with a computer relies in some way on an algorithm. Even the simplest application on a modern computer would not be possible without algorithms being utilized behind the scenes to manage memory and load data from the hard drive. Computer Algorithms help you to understand what you're doing, and how you can do it more efficiently and easily.

## II. COURSE MATERIAL PREPARATION

### A. Concept Map

After comprehensive study, contents of Computer Algorithm course are analyzed and the concept map is designed. The goal is to have the students be able to explain each part of the concept map and their reasoning behind the concepts. It helps students to understand and communicate a concept and its connections between examples and ideas.

Fig. 1 shows the concept map for Computer Algorithm course.

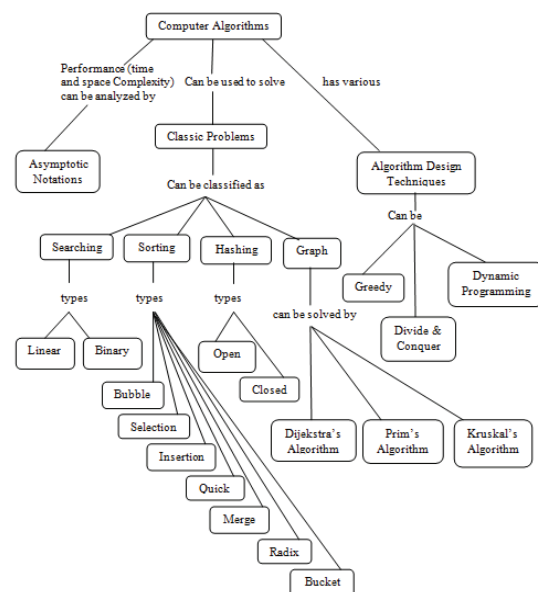


Fig. 1: Concept map for Computer Algorithm Course

### B. Course Learning Outcomes

Formal statements are framed which describe clearly what learners will know and be able to do at the end of the course.

After completion of the course, student will be able to:

- CLO 1: Understand and apply the mathematics needed for the analysis of algorithms.
- CLO 2: Identify, model, solve and develop algorithm for real life problems like change making problem, shortest path, and minimum spanning tree etc.
- CLO 3: Compute asymptotic notations to determine and analyse the performance/efficiency of algorithm and relate to the consumption of resources (time/space).
- CLO 4: Identify appropriate algorithm design strategies that are appropriate to a given contextual problem.
- CLO 5: Implement and compare various searching and sorting algorithms.
- CLO 6: Draw and test the flowcharts using some modern tool.

TABLE I. MAPPING OF COURSE LEARNING OUTCOMES SHOWS PREREQUISITE AMONG THEMSELVES.

CLOs	CLO1	CLO2	CLO3	CLO4	CLO5
CLO1					
CLO2	Y				
CLO3	Y	Y			
CLO4		Y	Y		
CLO5	Y	Y	Y	Y	Y
CLO6					

### C. Activity Plan

The activity plan is prepared to deepen the understanding of algorithms that will enhance the future problem solving skills. In this plan, various strategies for enhancing teaching and learning process are presented. It will make students understand the theories more deeply. It could effectively help

students to master the computer algorithm and apply them in different practical situations. Following activities are formulated under the activity plan.

- TPS (Think, Pair, Share)
- Poster Design
- Animation
- Demonstration
- Group Presentation
- Online Quiz
- Programming Test
- Debugging Test
- Use of Modern Tools
- Course Exit Survey

1) *TPS(Think, Pair, Share)*: Students worked on problem posed by teacher.

a) *Think*: Students think about what they know and come up with their own individual answer to the problem.

b) *Pair*: Each student is paired with another student. They share their thinking with each other.

c) *Share*: Students share their thinking with the entire class.

2) *Poster Design*: Groups of 3-4 students were formed and asked to choose the topics based on syllabus. They have selected the topic for poster and presented the poster.

3) *Animation*: A commonly used method for explaining algorithms is visualization. Adopting visual tools like animations, learning could be made much more effective. Some difficult and confusing topics were illustrated with the help of animation. For example, sorting techniques such as merge sort, quick sort, minimum spanning trees, graph algorithms like Prim's algorithm, Kruskal's algorithm, etc.

4) *Demonstration*: A practical explanation of how algorithm works is demonstrated. For example sorting algorithms such as insertion sort are demonstrated to the students with the help of example, graph algorithms such as Dijkstra's algorithm for finding shortest path from single source node to all other nodes in the graph.

5) *Group Presentations*: Groups of 3-4 students are formed. They have selected the topic and presented it in front of class which helps them to improve their self confidence and communication.

6) *Online Quiz*: Online quiz is conducted to test the knowledge of students on the moodle server.

7) *Programming Test*: Problem statement was given to the students. They have identified, developed and implemented the solution for it.

8) *Debugging Test*: Some non-executable code or pseudo code with missing statements was given to the students. They have converted it to the executable code.

rithm

9) *Use of Modern Tools- Rapter* : It is a flowchart-based programming environment, designed specifically to help students visualize their algorithms and avoid syntactic baggage. Using Rapter flowcharts are created and executed visually by tracing the execution through the flowchart. Required syntax is kept to a minimum. Students prefer flowcharts to express their algorithms, and are more successful in creating algorithms using Rapter than using a traditional language or drawing flowcharts without Rapter.

8) *Course Exit Survey*: Course exit survey is taken to evaluate efficiency of activity plan and the effectiveness of OBE approach for course preparation. It told how well the outcomes are achieved.

### III. IMPLEMENTATION

The above mentioned activity plan is included in:

- Course Plan
- Lab Plan

#### A. Assessment Methods

##### 1) Theory Course:

- a. ISE 20%
  - i. Online Quiz 10%
  - ii. Group Presentations 10%
- b. MSE 30%
- c. ESE 50%

##### 2) Practical Course:

- a. ISE 50%
  - i. CAS 40%
  - ii. Programming Test 10%
  - iii. Debugging Test 10%

#### B. How and When to Assess Outcomes

TABLE II shows the various learning and assessment strategies to achieve the particular course learning outcome.

TABLE II. ASSESSMENT OF COURSE LEARNING OUTCOME

Outcome	Description of question	When To asses (Learning Strategy)	How to asses (Assessment Strategy)
CLO 1: Understand and apply the mathematics needed for the analysis of algorithms.	Question will be posed which involves mathematical calculation.	1. Discussion on Board 2. During Class interaction 3. Class Assignment 4. Semester mid/end exam 5. During Practical's	Able to use the various mathematical concepts which are required to analyze the performance of algorithm.
CLO 2: Identify, model, solve and develop algorithm for real life problems like change making	Question will be posed to develop algorithms for some real life situation.	1. Discussion on Board 2. During Class interaction 3. Class Assignment 4. Semester mid/end exam	Able to solve provided graph problems.  Able to implement code that can find solution for above problems.

problem, shortest path, and minimum spanning tree etc.		5. During Practical's	
CLO 3: Compute asymptotic notations to determine and analyze the performance/efficiency of algorithm and relate to the consumption of resources (time/space)	Question will be posed to write asymptotic notation for time/space complexity for a given Program/algorithm.	1. Class Assignment 2. Practical's 3. Home Assignment 4. Semester mid/end exam	Able to write asymptotic notation for an algorithm.
CLO 4: Identify appropriate algorithm design strategies that are appropriate to a given contextual problem.	Question will be posed to understand all concepts, differences and should identify which algorithm can be used for that situation.	1. During Class interaction 2. Home Assignment 3. Semester mid/end exam 4. During Practical's	Able to demonstrate whole concepts with the help of case study.
CLO 5: Implement and compare various searching and sorting algorithms	Question will be posed to write and compare the performance of searching and sorting algorithms.	1. Class Assignment 2. Practical's 3. Home Assignment 4. Semester mid/end exam	Able to Implement executable code.
CLO 6: Draw and test the flowcharts using some modern tool.	Question will be posed to draw and test the flowchart for their correctness.	1. Class Assignment 2. During Practical's	Able to draw and execute flowcharts using Rapter.

#### C. Mapping of course learning outcomes with program outcomes of the institute

The following TABLE III shows the mapping of CLOs with POs of the Institute.

TABLE III. MAPPING OF CLOS WITH POS

	Program Outcomes
(a)	Graduates will be able to demonstrate knowledge of mathematics, science & engineering.
CLO 1:	Understand and apply the mathematics needed for the analysis of algorithms.
CLO 2:	Identify, model, solve and develop algorithm for real life problems like change making problem, shortest path, and minimum spanning tree etc.
CLO 3:	Compute asymptotic notations to determine and analyze the performance/efficiency of algorithm and relate to the consumption of resources (time/space).
CLO 5:	Implement and compare various searching and sorting algorithms.
	Knowledge of mathematics related to matrices, graphs, sets is required to achieve this outcome.
	Knowledge of engineering mathematics, geometric sequences and sums are

required to determine performance of an algorithm.	
(b)	Graduates will be able to identify, analyze, and formulate engineering and social problems using IT.
<p>CLO 2: Identify, model, solve and develop algorithm for real life problems like change making problem, shortest path, and minimum spanning tree etc</p> <p>CLO 4: Identify appropriate algorithm design strategies that are appropriate to a given contextual problem.</p> <p>CLO 5: Implement and compare various searching and sorting algorithms.</p> <p>Course includes identifying what type of algorithm technique is required for real world problem/situation.</p> <p>This course includes designing and solving complex problems like graphs and sorting problems.</p> <p>As a part of this course every algorithm that is designed should be implemented using C language.</p>	
(d)	Graduates will have an ability to design and develop software systems of varying complexity to meet the desired needs.
<p>CLO 2: Identify, model, solve and develop algorithm for real life problems like change making problem, shortest path, and minimum spanning tree etc.</p> <p>CLO 3: Compute asymptotic notations to determine and analyse the performance/efficiency of algorithm and relate to the consumption of resources (time/space)</p> <p>As a part of course students apply different algorithms to solve the same problem and analyze &amp; compare the performance of these algorithms like Prim's and Kruskal's algorithm to find minimum spanning tree.</p>	
(e)	Graduates will be able to make use of modern tools to evaluate and implement computer-based systems.
<p>CLO 6: Draw and test the flowcharts using some modern tool.</p> <p>Using Rapter flowcharts are created and executed visually by tracing the execution through the flowchart.</p>	
(g)	Graduates will be able to succeed in competitive examinations like GATE, GRE and various professional certifications in IT.
<p>CLO 1: Understand and apply the mathematics needed for the analysis of algorithms.</p> <p>CLO 2: Identify, model, solve and develop algorithm for real life problems like change making problem, shortest path, and minimum spanning tree etc.</p> <p>CLO 3: Compute asymptotic notations to determine and analyse the performance/efficiency of algorithm and relate to the consumption of resources (time/space).</p> <p>CLO 4: Identify appropriate algorithm design strategies that are appropriate to a given contextual problem.</p> <p>CLO 5: Implement and compare various searching and sorting algorithms.</p> <p>This course is very much useful for interview preparation, competitive exams preparation, and campus preparation. It focuses not only on theory but also on the analysis of problems to find the optimal solution.</p>	
(k)	Graduates will be able to analyze the impact of engineering solutions on individuals, society and environment.
<p>CLO 2: Identify, model, solve and develop algorithm for real life problems like change making problem, shortest path, and minimum spanning tree etc</p> <p>CLO 4: Identify appropriate algorithm design strategies that are appropriate to a given contextual problem.</p> <p>Travelling sales man problem, electrical network problems, shortest path problems, change making problem etc. are solved by the students in optimal way.</p> <p>Course includes identifying the impact of all possible solutions on individuals and/or society and picking the optimal solution suitable for real world problem/situation mentioned above.</p>	

#### IV. BENEFITS

Following benefits are seen after implementation:

##### 1) Tangible Benefits:

- Improves Result
- Students are able to design and analyze Algorithms

##### 2) Intangible Benefits:

- Subject Knowledge
- Problem Solving Skills
- Learning Skills
- Thinking Skills

#### V. RESULT ANALYSIS

The result of Mid SEM Exam (MSE) and End SEM Exam (ESE) is analyzed and compared with previous year End Sem Exam (ESE) result. As shown in the graph the results of ESE of year 2013-14 compared to the previous year has improved.

##### A. MSE Result Analysis

The TABLE IV represents total no of students appeared for the MSE exam and no. of students fall in different range.

TABLE IV. MSE RESULT ANALYSIS

MSE 74 Students					
0-5	6-10	11-15	16-20	21-25	26-30
0	03	08	24	23	16

Fig. 2 shows the graphical presentation of above MSE result data.

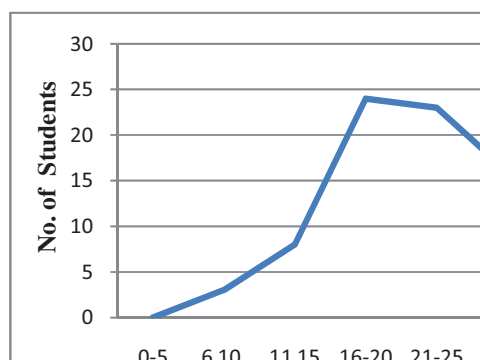


Fig. 2: The graphical Representation of MSE Results

##### B. ESE Result Analysis

The TABLE V represents total no of students appeared for the ESE exam and no. of students falls in different range.

TABLE V. ESE RESULT

ESE 74 Students						
0-39	40-49	50-59	60-69	70-79	80-89	90-100
04	09	13	19	16	12	01

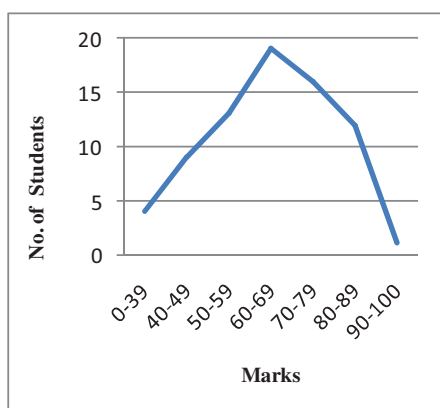


Fig. 3: The graphical Representation of ESE Results

Fig.3 shows the graphical presentation of ESE data of TABLE V.

### C. ESE Result Comparison between the year 2012-13 and 2013-14

The following Fig. 4 shows the comparison of ESE results for the year 2012-13 and 2014.

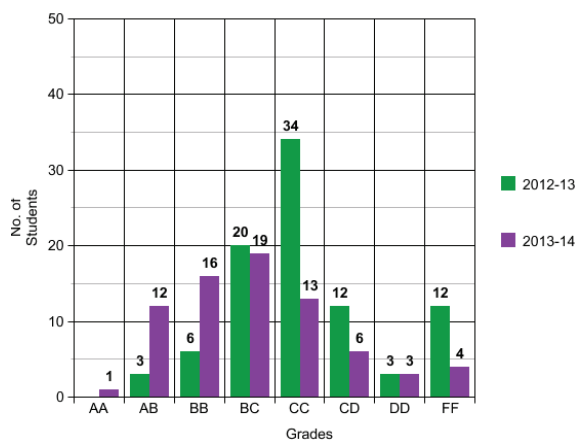


Fig. 4: The graphical Representation of ESE Results

### Conclusion

From the result analysis we can conclude that the results of MSE & ESE have been improved. Group activities motivate them, enhance their confidence level and increase the understanding of the subject matter. The feedback from students is also taken time to time to check the efficiency of every module. The course exit survey helped to find the attainment of course learning outcomes.

### References

- [1] Leigh Ann Sudol "Deepening Students' Understanding of Algorithms: Effects of Problem Context and Feedback Regarding Algorithmic Abstraction" June 2011.
- [2] Michael D. Byrnea, Richard Catramboneb, John T. Staskoc, "Evaluating animations as student aids in learning computer algorithms", June 1999; accepted 27 September 1999.
- [3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford, "Introduction to algorithms".