

# Structured Approach of Designing Data Structure and Algorithms Laboratory Experiments

**Moula Husain<sup>1</sup>, Somashekhar Patil<sup>2</sup>, Indira.B<sup>3</sup>, Meena S.M<sup>4</sup>, Narayan D.G<sup>5</sup>**

<sup>1,2,3,4,5</sup>B.V.B College of Engineering and Technology Hubli, india

<sup>1</sup>moulahusain@bvb.edu, <sup>2</sup>skpatil@bvb.edu, <sup>3</sup>indira\_bidari@bvb.edu, <sup>4</sup>msm@bvb.edu, <sup>5</sup>narayan\_dg@bvb.edu

**Abstract**-Data structures and algorithms play a major role in optimizing any non trivial software application. Learning and applying concepts of this course demands a great amount of analytical and logical thinking skills. Teaching this subject at early stages of Information Technology related engineering courses poses a great challenge to the Course Instructors. In this paper, we presented the experience of using structured approach of designing and assessing Data Structures and Algorithms Laboratory experiments. All our laboratory experiments and assignments are well designed and structured to achieve the desired ABET a-k learning outcome elements set for the course.

Our experimental case study examples, structured enquiry based experiments and student's feedback results prove the effectiveness of designing and adopting structured experiments in the laboratory course.

**Keywords**-ABET a-k Criteria, Bloom's Taxonomy; Bridge Course; Program outcomes; Performance Indicators;

## I. Introduction

Data Structures and Algorithm (DSA) is "Hard core" of Computer Science and Engineering fields. Time and space efficiency of any non trial software application can be optimized by applying appropriate data structures and algorithms. In most of the universities this subject is taught at early stages of the engineering course. Learning and applying such complex concepts at the early stages of engineering course to a software application requires a great deal of logical and analytical thinking skills.

The main objective of this course is to identify and apply suitable data structures and algorithms [14] [15]. Also students may need to make time and space complexity analysis by comparing it with other possible data structures and algorithms. In order to achieve effective learning of this course and meet desired learning outcomes, first time we have made an attempt to design our data structures and algorithms laboratory experiments according to increasing levels of complexity of the problem. Before the

---

**Moula Husain**

B.V.B College of Engineering and Technology Hubli, india  
moulahusain@bvb.edu

commencement of semester's laboratory term work, we conducted a "Bridge Course on C programming" to review basic concepts and enhance students C programming skills.

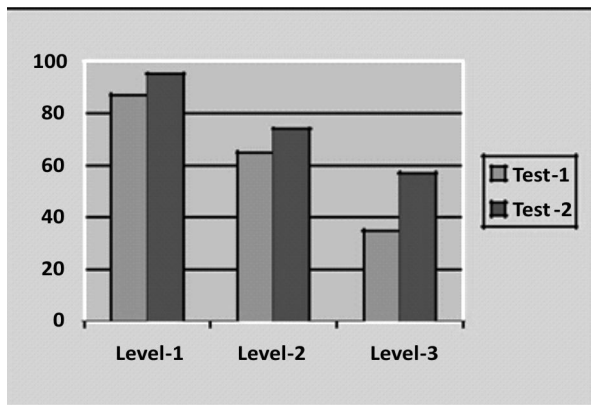


Figure1. Graph showing results of test1 and two drawn across Attainment of % and levels.

Figure 1 shows results of the test conducted before the start and after completion of bridge course. In order to analyze cognitive levels of the students, questions were framed according to the Bloom's

Taxonomy [1][2][3] learning levels. Surprisingly, analysis of graphs realized the fact that 80% of students were capable of solving L1 and L2 level questions. But more than 70% students failed to solve questions related to L3 category questions.

The major reason for this performance of students was lack of analytical and logical skills. To overcome the major lacuna, we designed all our laboratory experiments and structured enquiry assignments in increasing complexity levels by mapping with ABET a-k learning outcomes of the course. Benefits of adopting such a structured designed based approach are 1) Students can learn highly complex concepts in incremental complexity order 2) Helps course Instructor to identify intellectual levels at which individual students are capable of learning. 3) Develops high level thinking skills.

In the previous work [5] Sujatha, C et. al. have experimented an innovative approach of teaching data structures and algorithms subject first time on post graduate students and applied concepts of DSA for solving real world problems. Sarwar, S.M et. al.[6] have

adopted an approach of comparing results of the laboratory experiments with data structure's behavior.

In the paper by Lingfen Wang et al [7] Multilevel teaching methodology is adopted for effective teaching software engineering related courses by taking data structures and algorithms as case study. In many of the papers [8][9] algorithms related courses are taught by using visualization tools. In [10][11], redesigning and assessing of engineering program courses using ABET (Accreditation Board for Engineering and Technology) criteria[12][13]. In this paper we designed and assessed DSA laboratory course according to the program outcomes of ABET criteria.

The rest of the paper is organized as follows. Next section describes teaching methodology. In section 3, we discuss Bloom's taxonomy based laboratory experiment design. Section 4 describes the assessment methods followed in the course. In final sections, Results are demonstrated followed by conclusion and future scope.

## 2. Teaching Methodology

The main course objective is to identify and apply suitable data structures and algorithms. In order to achieve effective learning of the course, we designed our data structures and algorithms laboratory experiments according to increasing order of complexity level by considering Bloom's taxonomy. The course learning objectives (CLO) set for our DSA laboratory are as follows

- i. Develop the ability to troubleshoot and rectify syntax, logical and run time errors in C programs and use industry oriented coding conventions
- ii. Discuss the properties, operations, applications, strengths and weaknesses of the various data structures and their effect on algorithms
- iii. Identify, design and implement appropriate data structures like stack, queue, hash table, tree for a given problem
- iv. Implement the sorting, searching and graph algorithms by incorporating suitable data structures and perform efficiency analysis
- v. Integrate the various features of C and data structures in efficient implementation of given

application by selecting suitable algorithms

According to ABET a-k criteria, the following program outcome elements were addressed in the course and corresponding performance indicators were identified

**Table 1. Program Outcomes, outcome Element and Performance Indicator mapping**

<b>ISPO (a)</b>	<b>An Ability to apply knowledge of mathematics, computer science and engineering</b>
Outcome Element:	a-2. Ability to apply the knowledge of Computer science
a-2-B	Ability to apply the knowledge of algorithms, data structure and programming language
<b>ISPO (b)</b>	<b>An Ability to analyze a problem, identify and formulate the computing requirements appropriate to its solution.</b>
Outcome Element:	b-3. Ability to find appropriate solution and verify the correctness
b-3-A	Ability to chose appropriate data structure, algorithm, architecture
Outcome Element:	c-3. An Ability to design a computer based system, component or process to meet desired needs
<b>ISPO (c)</b>	<b>An Ability to design a computer based system, component or process to meet the desired needs within realistic constraints.</b>
c-3-A	Ability to develop a algorithm / functional module
c-3-B	Ability to implement the modules.
c-3-C	Ability to integrate the modules and verify the functionalities
<b>ISPO (g)</b>	<b>An Ability to communicate and present effectively in oral and written form.</b>
Outcome Element:	g-1. An Ability to document and communicate effectively g-2. An Ability to present effectively.
g-1-A	Ability to write clear and well organized project reports.
g-2-A	Ability to prepare presentation using visual aids.
g-2-B	Ability to deliver an effective presentation.

Considering the Course Learning Objectives, Program Outcomes, Performance Indicators and the results obtained in Bridge Course, we designed the experiments for the DSA laboratory. The first 7 lab slots belong to Demonstration and Exercise types. Each slot was split into 4 parts; every part is correlated to blooms taxonomy levels ranging from L1 to L3 and program outcome of the course. An example is illustrated for clarity of the proposed method:

onsidering the Course Learning Objectives, Program Outcomes, Performance Indicators and the results obtained in Bridge Course, we designed the experiments for the DSA laboratory. The first 7 lab slots belong to Demonstration and Exercise types. Each slot was split into 4 parts; every part is correlated to blooms taxonomy levels ranging from L1 to L3 and program outcome of the course. An example is illustrated for clarity of the proposed method:

**Table 2. Example illustrating L1 to L3 level of Blooms Taxonomy**

Sl.No	TASK	CLO, BL	PO, PI
1	Call any of the DSA Lab team faculty and answer the question, Why do we need Arrays? Define structure? [Marks: 10]	CLO 1 L1	a-2-B g-2-B
2	You are given with a code. The code has some errors and warnings. Debug all the errors and warnings. The code can be found in the file name "Matrix" [Marks: 20]	CLO 1 L2	a-2-B
3	Write a code which will generate the following warning message: Warning: arguments miss match for function [Marks: 20]	CLO 1 L2	b-3-A
4	Write a program for election process with following options: 1. Enrolment of voter with ID. • ID must be generated by program • ID's must start with 1111 (example: 1112, 1113, 1114...) 2. Vote for party. • Considering 3 party's in election • Ask for ID before voting. • Duplicate voting must not happen. 3. Calculating votes. • Display percentage of votes. 4. Exit. [Marks: 50]	CLO 2 L3	b-3-A c-3-B c-3-C

The lab slot was designed for practicing programs on arrays operating and enhancing knowledge on arrays and pointers. Task 1 is for recalling relevant knowledge addressing L1 level of Bloom's taxonomy and program outcomes "a" and "g". Task 2 emphasizes on programming skills addressing L2 and program outcome "a". This task is designed to build program debugging skills and enhancing it. Students are used to program in a traditional way wherein they make errors and debug it using trail and error methods but

more often, they don't understand why error had occurred. In Task 3, we tried the other way round wherein students are told to generate specified error statement, so that they understand the reason for occurrences of such errors. Hence it enhances the ability to verify the correctness of the program. It addresses L2 level of learning and program outcome "b". In task 4, students were told to implement a real world scenario of election process which helped in enhancing their problem solving skills. They had implemented the code with modular approach, integrating and verifying the functionalities. The last task addresses L3 level of learning and program outcomes "b" and "c".

The next 4 lab slots belong to Structured Enquiry which addressed L4 level of Blooms taxonomy and program outcomes "c".

An example is illustrated for clarity of the proposed method:

Question: Charlie and the Chocolate Factory  
Objective: Usage of linked lists

Scenario: Charlie, the new owner of chocolate factory after returning from 'Oompa-loompa' land has got a new recipe of making chocolates. Well, the old recipe was robbed out by a few workers and the chocolate factory had to shut down as every cat and dog in city started making the same kind of chocolates. Charlie had to shut his factory and make an adventurous expedition in search of new recipe.

That's when he ended up in Oompa- Loompa land. The Oompa-Loompas offered him with 2 new varieties in return of their favorite chow, beans. Now, one new secret recipe is shared only with you!

Here is the procedure:

First you need to buy the following:

- 200 KitKat
- 10 Silk
- 30 Bourneville
- 200 Wafers
- 30 Lint

Now you need to melt all in the same above order, one by one, whip it, grind it, bake it, freeze it and finally mould it to a dream fantasy chocolate. Your task is to do it all. Go to market buy all the chocolates required, melt them in the same order and prepare the new chocolate.

Your task is:

1. First buy all the required variety of chocolates. We may not get all in the same shop. If the desired quantity

is not available then we need to buy from next shop. Maintain a linked list structure:

```
struct choc_fantasy
{
    char choc_name[20]; int quantity;
    struct choc_fantasy * prev; struct choc_fantasy * next;
};
```

You will keep on calling the function "insert\_at\_end" until you buy all the required chocolates in specified quantity. The required chocolates can be purchased in any order.

2. Melt and prepare fantasy chocolate Now melting process has to happen exactly in the same way as said. You will search for kitkat in linked list and delete the nodes (it could be one node having 200 quantities or split up in any number of nodes). Print appropriate message after adding each ingredient.

3. The print the message that "Secret Recipe Chocolate is now ready" and exit the program.

Structured enquiry was designed to emphasize on analyzing capability of students. To achieve this we framed scenario based questions addressing L4 level of Blooms taxonomy. In the above mentioned example, we have taken a movie scenario so that students get fascinated in solving them and enhanced there analyzing and applying capabilities.

### 3. Assessment Method

Ours is an autonomous college, here we have continuous internal evaluation (CIE) of laboratory. The allotted CIE marks are 80 and we have divided it into two parts: part A (Demonstration and Exercise) which carries 30% of CIE marks and part B (structured Enquiry) carries 50% of marks. The other 20% of marks is evaluated in Semester End Exam (SEE) as shown in Table 3.

**Table 3: Students Assessment through CIE (80%) + SEE (20%)**

Continuous Internal Evaluation (80%)	Assessment	No.Of. Lab slots	Weightage in Marks
	Demonstration	03	10
	Exercise	04	20
	Structured Enquiry	04	50
Semester End Examination (20%)	CIE Total		80
	SEE Experiment		20
	Total		100

#### 4. Percieved Benefits

At the end of semester, student's feedback about the course conveyance was taken. The questions asked were based on the learning ability, analysis of code, debugging and programing skills.

The questions asked in feedback are as listed below:

1. Learning how to choose an appropriate data structure and algorithm.
2. Ability to analyze the code for any given problem statement with respect to time and space complexity.
3. Capability of finding bugs in a given code and rectifying them.
4. Improving problem solving ability and programing skill.

We analyzed the effectiveness of practicing structured based approach by comparing semester - end DSA

laboratory exam results of the two consecutive academeffectiveness of adopting structured based approach into our teaching years 2013-14 and 2012-13.

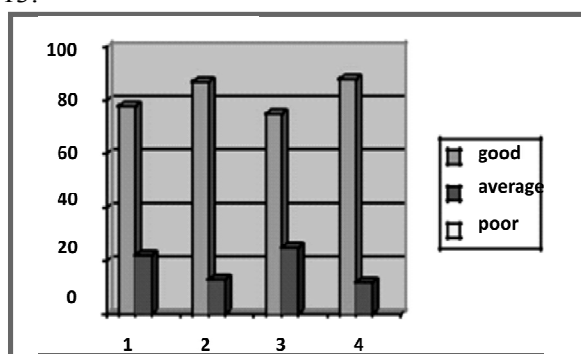


Figure2: Students feedback analysis graph

practices. In future, we plan to use such teaching practices for other programming related courses.

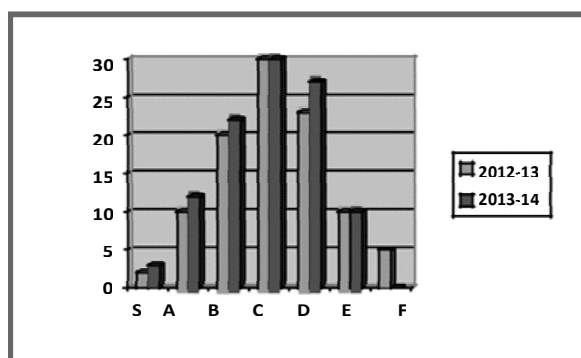


Figure 3: Comparison of semester end DSA laboratory exam results of two academic years and results (2012-13 & 2013-14)

Figure 3 reveals the fact of improvement in the performance of the students in the semester end exams. More significantly, this has helped the students to understand and analyze concepts of algorithms and data structures. At the end of this course, students are in a position to identify suitable data structure and associated algorithm for any given problem specification. Further, they were able to analyze time and space complexity of the program by applying various combination of data structures and algorithms.

#### 5. Conclusion

In this paper, we have presented structured approach of designing and conducting DSA laboratory experiments. The laboratory experiments are well structured and designed according the Bloom's taxonomy levels and ABET a-k based program outcome elements. We described the process of solving a complex problem by breaking it in increasing order of learning levels. Also we have demonstrated the process we used for assessing these outcomes using performance indicators. Our course design methodology, assessment criteria, student's feedback and semester end exam results prove the

#### References

- [1] Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; Krathwohl, D. R. (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: David McKay Company.
- [2] Orlich, Donald; Harder, Robert; Callahan, Richard; Trevisan, Michael; Brown, Abbie (2004). Teaching strategies: a guide to effective instruction (7th ed.). Houghton Mifflin. ISBN 978-0-6182-9999-7.
- [3] Colin G. Johnson and Ursula Fuller. 2006. Is Bloom's taxonomy appropriate for computer science?. In Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006(Baltic Sea '06). ACM, New York, NY, USA, 120-123.
- [4] Spivey, G., "A Taxonomy for learning, teaching, and assessing Digital Logic Design", Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Oppurtunities without passports, 2007, FIE '07. 37th Annual, vol., no., pp. F4G-9, F4G-14, 10-13 Oct. 2007
- [5] Sujatha, C.; Jayalaxmi, G.N.; Suvarna, G.K., "An innovative approach carried out in data

- structures and algorithms lab," Engineering Education: Innovative Practices and Future Trends (AICERA), 2012 IEEE International Conference on , vol., no., pp.1,4, 19-21 July 2012
- [6] Sarwar, S.M.; Parks, E.E.; Sarwar, S.A., "Laboratory exercises for practical performance of algorithms and data structures," Education, IEEE Transactions on , vol.39, no.4, pp.526,531, Nov 1996
- [7] Lingfen Wang; Xizuo Li; Wei Li; Haiyu Song, "ZPD-Based Multi-level Teaching Pattern in Software Engineering--A Case Study of Data
- [8] Structure and Algorithm Course," Education Technology and Training, 2009. ETT '09. Second International Conference on , vol., no., pp.7,9, 13-14 Dec. 2009
- [9] Cardim, G.P.; Marcal, I.; de Sousa, C.M.; de Campos, D.L.; Marin, C.H.V.; do Carmo, A.F.C.; Toledo, D.F.; Saito, A.; Correia, R.C.M.; Garcia, R.E., "Teaching and learning data structures supported by computers: An experiment using CADILAG tool," Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on , vol., no., pp.1,5, 20-23 June 2012
- [10] Zulhisam, M.; Rias, R.M., "Modeling Binary Tree Algorithm in 3-D Interactive Visualization: A Prototype," Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on , vol., no., pp.525,528, 8-10 Feb. 2012
- [11] C.R.; Gurocak, H., "Assessment methods for the upcoming ABET accreditation criteria for computer science programs," Frontiers in Education Conference, 2008. FIE 2008. 38th Annual , vol., no., pp.S4G-6,S4G-11, 22-25 Oct. 2008
- [12] Wellington, C., "Designing an ABET accreditable computer science bachelor's program within the constraints of a teaching university," Frontiers in Education, 2004. FIE 2004. 34th Annual , vol., no., pp.T3G/18,T3G/23 Vol. 1, 20-23 Oct. 2004
- [13] A B E T . 1 5 t h O c t o b e r 2 0 1 4 . <http://www.abet.org/etac-criteria-2014-2015/>
- [14] Aaron M. Tenenbaum, Yedidiah Langsam, and Moshe J. Augenstein. 1990. Data Structures Using C. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [15] Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasckaran. 1996. Computer Algorithms: C++. W. H. Freeman & Co., New York, NY, USA.
- [16] Bloom's taxonomy 25th October 2014 <> [http://ww2.odu.edu/educ/roverbau/Bloom/blooms\\_taxonomy.htm](http://ww2.odu.edu/educ/roverbau/Bloom/blooms_taxonomy.htm)
- [17] Bloom's taxonomy 25th October 2014 <> <http://cft.vanderbilt.edu/guides-subpages/blooms-taxonomy/>