# Comprehensive Curriculum of Programming for Engineers, its Teaching Models and in-Lab Monitoring Technique

Mohit Satoskar[1], Sagar Mali, *Member, IAENG*

**Abstract – Issues related to the curriculum design of C programming language is analysed. Aspects and necessity of programming for engineers is discussed. The enhanced outlay of programming for engineers and generating the curriculum is presented. Relativity and dependency briefly highlighted. A superlative teaching model of computer languages is discussed along with its principles. In-Lab monitoring technique is developed successfully using client-server architecture which ensures learner live workshop progress. It is shown how the programming for engineer's curriculum design and techniques help to create highest quality outcome**.

**Keywords: programming, curriculum design, relativity and dependency, computer languages, teaching model, in-lab monitoring technique, client-server architecture.**

## I. Introduction

The pace of technological progress and evolution in programming languages escalated rapidly during the opening decade of the 21st Century and these dynamics were immediately channelled from Procedure Oriented Programming (POP), Object Based Programming (OBP) and now Object Oriented Programming (OOP).

The supremacy of programming in areas ranging from miniscale innovations to aviation, military or telecommunication, all resulted in the computing revolution. As a result, programming gained an unparalleled role in varsity education.

With programming becoming nothing short of showbiz, academia and university curriculum started including and applying the basic structural content. Programming curriculum remained low-key on many counts, especially in curriculum design and teaching. Hence to teach it successfully, we need to structure our techniques to realistic objective mode [1] [2] [3].

Manuscript received October 4, 2014; revised November 13, 2014

[1]Mohit N. Satoskar is a research associate with the Department of Computer Science and Information Technology at Ramrao Adik Institute of Technology, Mumbai - INDIA. e-mail: mohitsatoskar@gmail.com

Sagar Mali is in Computer Programming and System Software Design area of Computer Science at Kolhapur, Maharashtra, INDIA. email: sagaryash@gmail.com

However, the programmer or the student is completely aware of the so called full learning lifecycle of C programming as it is a necessity for computing, electronics and engineering student. As a matter of fact, C is the only language which can be used in multiple fields, areas which are not limited to computing only. Thus, it acts as a foundation.

The capsule of C programming contains combination of science, technology, engineering and maths (STEM). It was in 1972 that the first C- Programming Language was introduced, and since then C has dominated the world of programming languages [6] [8]. At present, the courses are mostly traditional and focus on limited area of C within the academic world (at least in the UK and India). It is seen that C language played a leading role in revolution of programming as many programming languages borrowed their base structure directly or indirectly from C [6] [7] [8]. However, there are several add-on concepts which will help to have a comprehensive study of C programming.

Typically, this paper presents the major 'override' contents which can shape curriculum of C for higher educational institutes. The objective of this paper is to strengthen the Curriculum of Programming for Engineers, and its creative teaching.

## II. Vision And Mission

So far, C programming is widely used in its basic structural format, however, in academics the base structure should be of comprehensive nature. There are many aspects of the current curriculum that need instant reformation in order that the new curriculum is effectively implemented. The following areas should be looked into with diligence.

- As of now, only primary content of C is being taught during higher studies.
- As the basic structure is being used in various other advanced languages, it is required to expand the margins of curriculum of the same.
- The required body of curriculum must be made as efficient as possible.
- This can broaden the aspects of C which are still unhandled by students.

- Finally, we need to turn our approach from a collection of teaching methods into a systematic methodology.

## III. Analysis, Relativity And Dependency Of C

Most educational institutes highlight the single set of primary segment in C curriculum in higher educational studies. Figure 1 highlights the traditional curriculum of C programming. However, C completely relies on procedures as the figure shows the step by step learning contents of C programming for Engineers. Due to these limitations, science consumer (student) is unaware of the rest of the contents in C. The following flow highlights the typical curriculum which is taught in major higher educational institutes. Pointing to this, it is required to pop in *Graphics, Hardware and Memory Management, File and Directory Management*, separately at a deeper level [6] [7] [8].

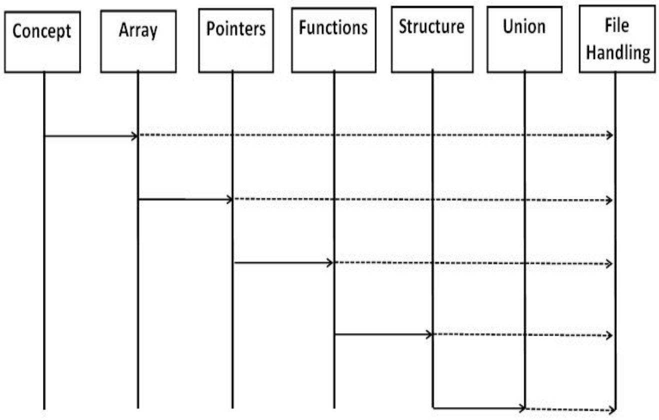


*Fig 1: Typical Primary Curriculum of C*

## IV. Comprehensive Structure Of Programming For Engineers Curriculum

So far many educational institutes are imparting typical structure of C programming language, shown in figure 1. However there are major sundry contents which need to be included in academic curriculum [8]. Initially, Figure 1 indicates primary base structure. Comprehensive Structure of Programming for engineers is divided in 4 substantial parts: Primary Basic C, Graphics, Hardware and Memory Management and File/Directory Management. The lessons go in stages and a learner has to complete each one before the next. Figure 2 indicates the comprehensive structure of programming curriculum

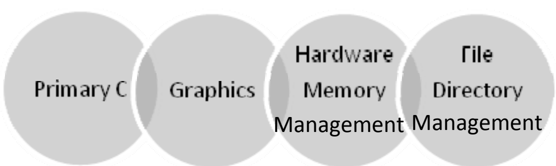


*Fig 2: Comprehensive Curriculum for Programming for Engineers*

In any case, comprehensive curriculum should not be seen as a standalone 'Programming Language Course' but as a part of an overall teaching toolbox or rather it should be adopted as a language instead of a course. *Graphics, Hardware and Memory Management, File and Directory Management* are the most crucial areas of learning and teaching. It is directly related to Engineering Problem Solving, Program testing with optimisation, Signal and Image processing, Data Acquisition and Control Applications[6] [7] [8]. Thus, it is not limited only to 'Computer Science'; rather it is the 'backbone' of Computing, Electronics and Telecommunication branches.

## V. Principals Of Teaching Model

Teaching and learning are two sides of the same coin. They are interdependent processes, thus, for an outcome of excellence both have to be equally strong. A vital first step is to analyse the problem and then to understand the logic. Logic is the foremost key element in the area of programming. Typically '*Teaching for Programming Language'* consists of 5 parts, *Planning, Conjoining, Apprising, Briefing and Exercising* [2] [4].

## VI. Subjective Teaching Model

There are several different ways in which a teacher can develop student's programming language skills. For example, reading a textbook, analysing and logical self generated thinking, or carrying out practical experiments. Generally, we need all of this working together so that they complement one-another. To achieve this goal we need '*Teaching for Programming Language model'*. Three very complex entities are involved in this. The most generalised model shown in Figure 3.

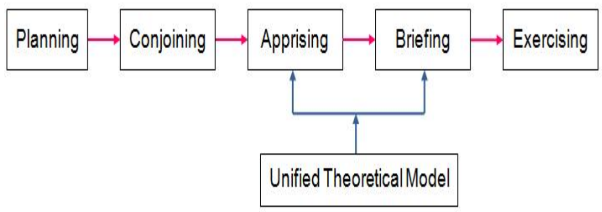


*Fig3: Generalised Traditional Teaching Model*

The model shown in Figure 3 represents a traditional teaching model, as its inception is from planning phase in which layout is planned of specific topic. Conjoining refers to gathering information, which the teacher is supposed to explain to the learner. Apprising phase highlights primary key points, called as structural base line for the whole subject. The subject has different shades called as Contents. However, each content has its own theory which differs from Base theory. The Briefing phase will explain in detail the theory of each content. Therefore, In *Subjective Teaching Model*, Unified Theoretical model composed of Apprising phase (Structural base theory) and

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Briefing phase (Content wise theory). Continuing with traditional model, Exercising phase (Practical) stands at the last position.

## VII. OBJECTIVE TEACHING MODEL

As the teaching model varies from subject to subject, programming languages concepts are directly related to the real world, thus it is required to teach practical model prior to briefing phase so that learner can understand it more smoothly. Thus it is needed to reform the process of traditional teaching by minor yet substantial reformation.

Though the theory provides practical as a base, there is a lot of difference when it comes to the realistic practical work one has to do and the theory (of the same subject) one has learnt, resulting in a major struggle.

To sort out this conflict, exercising phase (practical model) will develop a learner's critical thinking ability, indicating the logic required for problem solving. Figure 4 highlights the modified teaching model of programming for engineers. However, this model can help to assess learner improvement in future or logical ability depending upon workshop or realistic programming usability.

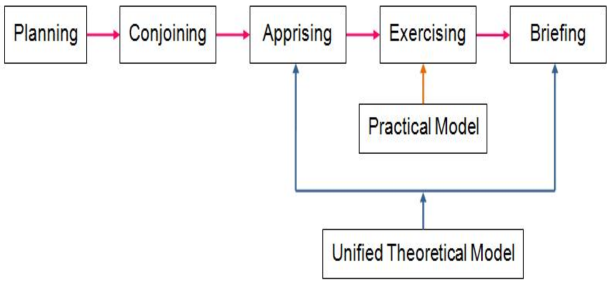


*Fig 4: Modified Teaching Model of Programming for Engineers*

The above model is an integral part of teaching and learning based on practical objects. Under this tagline, teaching the practical model first and then explaining the relevant theory will build the logical core called as '*Objective Teaching Model'*.

Considering the spoken language terminology, there are several different ways spoken language can be taught, the primary one's being visual, auditory, tactile or kinaesthetic [9]. But being a 'programming language' above primary ways do not apply to this. Programming Languages can be taught through visual, auditory and practical means. In C programming, one given problem can be solved through many ways but the skill is to choose the optimum path. Programming Language can be taught more effectively through *Object Based Teaching*.

The more the existence of practical based teaching, the better logical development takes place. This is usually categorised as 'integrative'- the desire to teach programming practically which focuses on various paths to tackle and solve the problems; then it leads to briefing phase in which learner can analyse the working process of the same which will make the learner conceptually and logically strong. Learner will understand the reason behind using a specific technique or method, and how it can reduce code in the stipulated time-frame in the real world, and so on. It is hard to put forward the technique of code reduction along with choosing the optimum path to surpass the problem in programming through generalised teaching model shown in Figure 3. Thus, actual working structure can be more legible to learner after the teaching model is modified, as highlighted in Figure 4.

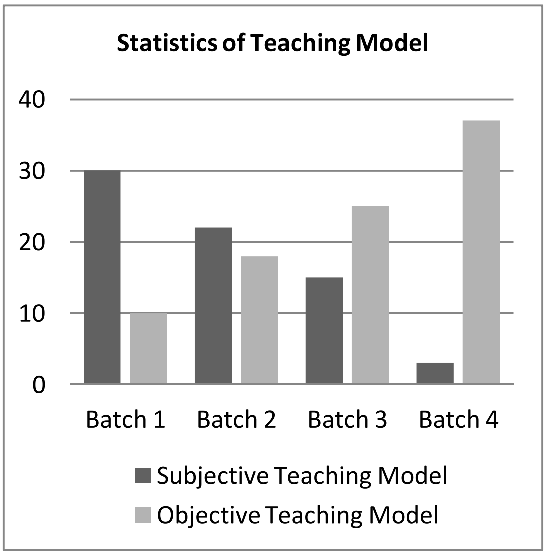


*Fig 5: Statistics of Teaching Models*

Figure 5 depicts the outcome of the study conducted in various batches of the programming for engineers (C programming language). The result, as a whole, looks very promising in the programming for engineering module. Both of the models have been implemented at AVIT academy for several C programming batches. First dark area refers to student's preference for *Subjective Teaching Model* and second light shaded area shows student's preference *Objective Teaching Model*. Gradually, it is seen that objective teaching model gives much more promising results comparing to subjective teaching.

## VIII. IN-LAB MONITORING TECHNIQUE

It is important to design and develop coherent approaches to planning, learning, teaching and assessment and to sharing information about progress and achievements.The majority of programming languages teaching will be based on workshops and then relevant theory. In traditional education system, there exist some loopholes in terms of the students' live-lab-progress. Lab monitoring technique is based on client-server architecture, which will act as a thread and capture the screenshot of students' system in such a way that the teacher can check what the student has been doing in lab. Thus, it will help the teacher to analyse students' live progress in laboratory practical workshops.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Following Algorithm shows the process of in-lab monitoring technique

1 Set the client- server architecture.
2 Start the server and so that it automatically detects connected clients.
3 After certain specified time (user defined time interval) server captures the screenshot of active machines.
4 Stores all the screenshots into specific destination on the server in user defined standard image format.
5 Through this teacher can check individual learner progress during workshop.

The above algorithm uses client- server architecture. It captures screenshot of active machines in user defined time intervals operating in milliseconds. And automatically saves the captured images in server side. The image format can be .jpeg , .png or .gif format. This application is purely based on platform independent approach.

## IX. Trial Application

A trial application of In-Lab monitoring technique is tested at AVIT solutions. Figure 6 shows the basic trial application model.

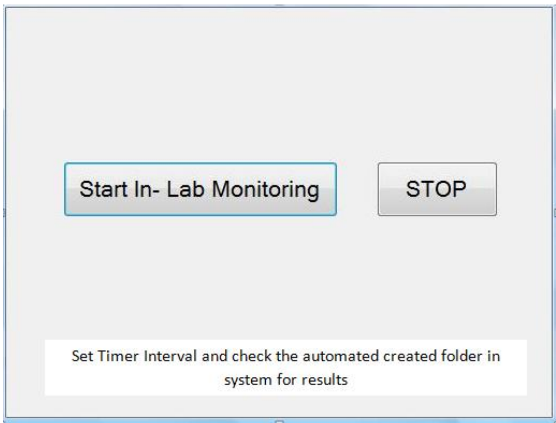


*Fig 6:In- Lab Monitoring Technique Trial Application*

The application can be used in schools, colleges as well as in the universities. It gives promising results directly to the responsible person. This application captures live progress as a .jpeg image of the students' or clients' system. Moreover, it can be used for employee appraisal.

## X. Conclusions

Initially, vision and mission of this research has been briefly defined. In this paper; we have described a comprehensive structure of programming for the engineering curriculum; Modified teaching model in terms of reforming phases of Subjective teaching and Objective teaching model. *Teaching for Programming Language* phases were analysed using *Planning, Conjoining, Apprising, Briefing and Exercising.* Considering 'exercising phase' prior to 'briefing' was effective to understand the logical aspect and helps to develop the learner's critical thinking capacity.

*In-Lab monitoring technique* has been introduced to check learner's lab based workshop progress.

This structure is committed to imparting cutting-edge curriculum for learners of Computing, Electronics and related fields. This paper provides the highest quality curriculum for the learners of science with different skills.

***Acknowledgements***- The author gratefully acknowledges the support from Miss. Anuprita Satoskar during the undertaken study. Mohit N. Satoskar would like to thank Dr. Vijay D. Patil for the continued support in this endeavour. This project will not be completed without your continuous motivation. Also, the authors wish to thank Dr. Ramesh Vasappanavara and Mrs. Tejaswinee Suhas Deshpande for helpful discussions. In addition, this work was supported in part by RAIT, D.Y.Patil Group, Mumbai-India.

## References

[1] L. Wittek, L. Habib, "Quality Teaching and Learning as Practice Within Different Disciplinary Discourses", International Journal of Teaching and Learning in Higher Education. 2013. ISSN: 1812-9129.
[2] King, "C Programming: A Modern Approach," W.W. Nortan and Co; 2nd Edition, May 2008. ISBN: 0393979504.
[3] "Engineering and Technology (E & T) Magazine- IET", Vol 9, July 2014. ISSN: 1750-9637.
[4] A. Rath, A. Jagdev, S. Swain, "Programming in C; Second Edition, Scitech Publications. ISBN: 9788183714419.
[5] Mohit Satoskar, "An Enhanced Evaluation Approach: Computer Science and its Framework", The International MultiConference of Engineers and Computer Scientist, Hong Kong 2014. ISBN: 987-988-19252-5-1.
[6] M. Charnine, V. Protasov, "Optimum Automated Method for Collaborative Development of University Curricula 2014.
[7] Z. Ren, C. Yen, G. Liu, "Application and Research of C Language Programming Examination System Based on B/S", Third International Synopsys on Information Processing (ISIP), Qingdao, 2010. ISBN: 978-1-4244-8627-4
[8] M. Robbins, I. Schagaev, J. Yip, "Effective Teaching in Theoretical IT Modules", 33rd Annual Frontiers in Education Conference", (FIE'03)2003, http://doi.ieeecomputersociety.org/10.1109/FIE.2003.1263280
[9] Ramsden, P. "Learning To Teach in Higher Education, London, Routledge, 1992.
[10] K. Oza. "Applying Data Mining for Framing of Computer Science Curriculum", Proceedings of The International Engineering and Technology Education Conference (IETEC'13), Ho Chi Minh City, Vietnam