Developing Logic Building, Problem Solving, and Debugging Programming Skills Among Students

Dr. Sandeep A. Thorat[1], Durgesh P. Kshirsagar[2]

[1]Computer Science Department, RIT Rajaramnagar, Sangli, Maharashtra.
[2]Computer Science Department, RIT Rajaramnagar, Sangli, Maharashtra.
[1]sandip.thorat@ritindia.edu
[2]durgesh.khirsagar@ritindia.edu

Abstract: Computer programming course has become integral part of first year curriculum in all engineering disciplines. The effectiveness in delivery of this Course largely impacts on student's curiosity in the field of programming and software development. Many students lose their interest in the coding at very early stage of their professional career if this Course is not taught effectively. The proposed work aims to improve academic delivery of the computer programming course.

This work identifies loopholes in existing method of teaching computer programming course. Further we attempt to address these identified problems with different approaches. The approach focuses on development of logic building, problem solving, and debugging skills among the students. It is observed that if these skills are addressed properly then realistic improvement in programming skills of students can be achieved. We observed improvements in academic results of computer programming course after implementation of the proposed method. The approach proposed here can make learning computer programming a joyful experience for the students.

Keywords: First Year engineering, Computer programming, logic building, problem solving, debugging, effective academic delivery.

1. Introduction and Motivation

A. Introduction

Information Technology (IT) has become integral part of various types of industries and businesses. IT has applications in various industries like manufacturing, constructions, electric power, telecommunication, automobile, chemical etc. Further, IT is integral part of various scientific and research jobs. All these industries are recruiting employees who are having at least basic programming skills. Due to this industrial significance, computer programming course has become integral part of first year curriculum across various engineering disciplines. AICTE model curriculum for first year engineering also recommends this Course to be taught in first year.

The syllabus of computer programming course generally covers C programming language. C is considered as fundamental programming language. It is believed that a candidate who is expert in C programming can acquire expertise in any other programming language or technology quickly. This is particularly necessary in IT industry wherein technologies are changing rapidly and there has to be quick migration of new technologies time to time. Following are Course Learning Outcomes of first year computer programming course given by AICTE [2].

The student will learn

1. To formulate simple algorithms for arithmetic and logical problems.
2. To translate the algorithms to programs (in C language).
3. To test and execute the programs and correct syntax and logical errors.
4. To implement conditional branching, iteration and recursion.

In our Institute many students are from rural background. These students are having less exposure to computer systems before they join engineering studies. Many of them faces various issues while handling computer systems in effective way. To address this issue, we made few revisions in course syllabus. Besides the C programming we included following topics in the syllabus of this Course are – Introduction to Computer Hardware and System Software's (Operating System basic administration).

Making students expert in the programming is an ultimate course learning outcome of this course. Following are revised Course Learning outcomes for the Computer Programming course in our Institute.

1. To be able to operate computer systems in effective way.
2. To formulate Algorithm for given problem statement.
3. To translate given algorithm in C program.
4. To apply advanced C topics like Arrays, Pointers, and Structures.
5. To debug given code for identifying logical errors in the program.

The syllabus of this course is inspired from Dennis Ritchie [11] and E. Balaguruswami [12] text books.

The rest of paper is organized as follows. Section 2 discusses literature survey related with this topic. Section 3 presents motivation behind proposed approach. In section 4 the details of proposed approach to improve academic delivery of computer programming course are given.

Section 5 discusses academic results observed after applying the proposed method. Section 6 concludes the work.

2. Literature Survey

The difficulties faced by first year students while learning computer programming course are discussed in details by (P. Chenna Reddy, 2015). Depending on observations in this study the author has also given guidelines to improve methodology for teaching programming course. Various challenges in teaching computer programming courses are also discussed in depth by (S. Kannan, 2018).

(Abhiram Ranade, 2016) emphasized on implementation of interesting programs from various disciplines like science, engineering, operations research. The author mentioned that this approach is useful to make deliver of computer programming course more interesting. (Sunitha P., 2015) also discussed importance of programming assignments to improve student’s involvement in computer programming course. (Uma Boregowda, 2017) suggested personal motoring of weak learning students can improve overall academic results of computer programming course to a great extent.

(Varsha T. Lokare, 2018) discussed importance of logic development skill in delivery of various programming courses. The author discussed different methods for improving logical skills of students. (Jayashree Patil, 2019) encouraged use of ICT tools in academic delivery for logic development activities.

3. Motivation Behind Proposed Approach

The above literature survey helped us a lot to understand various issues related with academic delivery of computer programming course and possible methods to overcome these issues. Besides this literature survey we also conducted our own survey to understand the exact requirements related with this. As part of our own survey we informally interacted with Second Year students and faculty members.

We informally interacted with Second Year students of various engineering disciplines to understand possible challenges in learning computer programming course. We also interacted with teachers who taught these students to know their perception about the various issues. Following issues were cited by most of the students and faculty members during this interaction –

1. Anxiety about Computer Systems - Few students are lagging basic computer handling skills. This includes starting of computer, installing new software on computer, accessing Internet etc. Such students have fear and anxiety about the Computers. So special efforts are required to make these students comfortable with computer systems.

2. First Time Programmers - Most of the students are learning computer programming language first time. They are unable to cast themselves in role of a programmer and they just expect that computer will do required things for them automatically.

3. Casual Approach Towards Syntax - C programming is case sensitive and strictly type checked programming language. The syntax of C programming needs to be followed strictly. Students tend to over-look these issues; which results into generating many compiler errors.

4. Logic Development - To be a good programmer student need to be good in logic development. Here for given problem a computer algorithm is to be designed. The algorithm writing is generally ignored by the students and they hop in to start implementation of the program itself. Focused efforts are required to build logic development ability among the students.

5. Practice Makes Better Programmer - Students need more practice activities to get comfortable with programming environment. Many students do not implement sufficient number of programs. More practice activities help to get expertise in the programming.

6. Overcome Logical Errors – When students are preparing their own logic to solve given problem statements many logical errors remains un-addressed in the solutions. Students are unable to identify these errors and they face difficulties while overcoming such errors.

We used various approached to address issues listed above. Following sections explains these approached in detail.

4. Proposed Method

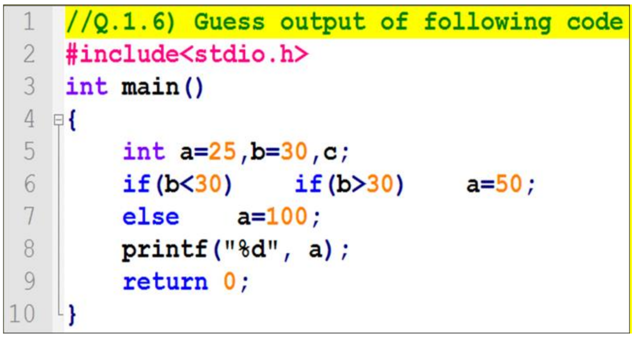
In this section we discuss proposed approach for academic delivery of computer programming course in details. There are three major requirements for effective delivery of programming courses. Mastering the concepts and syntax of a programming language is an elementary step to be a good programmer. Another important requirement is student should be able to write computer algorithm for any given problem statement. Last requirement is student should be able to identify and remove logical errors in the code (P. Chenna Reddy, 2015). These three requirements can be addressed effectively by imbibing problem solving, logic building, and debugging skills among the students. The work discusses importance of each of these skills and presents action plan for building them among students.

To incorporate above skills the proposed method continuously put different types of challenges in front of students. These challenges are uniquely designed for each topic in the Course with increasing difficulty level. In academic delivery basic concepts of each topic are taught using 50% of time. Advanced concepts of each topic are taught by putting different challenges in front of student and asking them to overcome the given challenge by applying taught concepts. This method creates interest in programming among students and also improves students’ active participations in the classroom activity.

We addressed various issues listed in Section 3 during academic delivery of computer programming using different approaches. Table 1 gives details of issue addressed and method used for the same.

Table 1. Proposed Approach for Academic Delivery of Computer Programming Course

Sr.	**Issue Addressed**	**Method / Approach**
1	Anxiety about Computer Systems	• Demonstration of basic computer operations in front of students • Small activities assigned to students which are to be done during Lab sessions. E.g. Change password of User, add folder, rename folder, check free space in drive, install Software, Change date & time etc.
2	First Time Programmers	• Many examples used to bring clarity about following topics - What computers can do and limitations of computer systems. - Purpose of computer programming and general features of computer programming language
3	Casual Approach Towards Syntax	• Topic-wise short code challenges prepared and used during academic delivery. • Short codes as shown in figure 1 used to have good understanding of C syntax and related concepts. • Students were challenged using such coding puzzles. The polls conducted using PollsEveryWhere tool. This improved student's interactivity in the classroom.
4	Logic development and More Practice Activities	• Asking student to write Algorithm using pen & paper. • Implement strict policy as no access to computer systems unless Algorithm is written properly by the student. • Using topic-wise unique problem statements in increasing difficulty level. Sample examples of topic-wise programming statements is shown in figure 2.
5	Overcome Logical Errors	• Demonstration of how debugging is done in computer programs. • Generating code with bugs and asking students to remove that bug. Sample example code with logical error is shown in figure 3. • Challenging students to do the debugging in shortest amount of time.

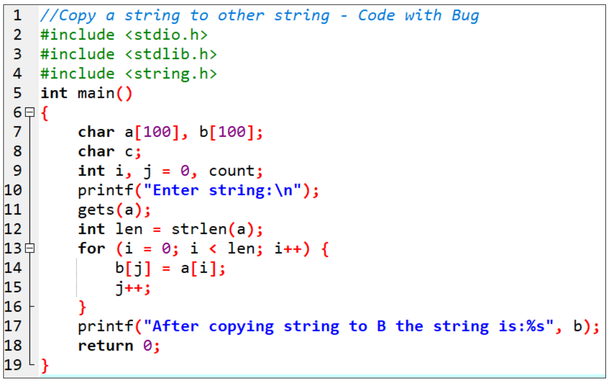


```
//Q.1.6) Guess output of following code
#include<stdio.h>
int main()
{
    int a=25,b=30,c;
    if(b<30)    if(b>30)    a=50;
    else    a=100;
    printf("%d", a);
    return 0;
}
```

Figure 1: Sample Example of Short Code Challenge

Implement following C codes using strings.	
1	Find length of given string
2	Find string length using recursion
3	Check if given two strings are equal
4	Copy one string into another string
5	Reverse given string
6	Reverse given string (using recursion)
7	Delete all occurrences of given character from the string.

Figure 2: Sample Topic-wise Problem Statements



```
//Copy a string to other string - Code with Bug
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char a[100], b[100];
    char c;
    int i, j = 0, count;
    printf("Enter string:\n");
    gets(a);
    int len = strlen(a);
    for (i = 0; i < len; i++) {
        b[j] = a[i];
        j++;
    }
    printf("After copying string to B the string is:%s", b);
    return 0;
}
```

Figure 3: Sample Example of Code with Logical Error (Bug)

5. Academic Results

This section discusses assessment method used for computer programming course, course results and course learning outcomes attainment.

A. Assessment Method

The assessment method adopted by us is inspired from the one proposed by Dr. Paresh, 2020 [10]. In the First-Year curriculum computer programming is purely laboratory course. Hence this course doesn't have any written examination. Also, there is no end semester examination for this Course. Hence, we need to do In-semester evaluation for 100% weightage. We used four different components during the In-Semester Evaluations. The details of these components are as follows –

i. Performing Lab Experiments

Students need to perform 10 experiments during laboratory hours. The student's performance during these experiments is assessed using Rubrics sheet. The weightage assigned to this activity is 50% in the total evaluation process.

ii. C MCQ Tests

MCQ Tests help to assess student's knowledge level about C programming syntax and overall conceptual understanding. We conducted 3 MCQ tests each of 25 marks. The weightage assigned to this activity is 20% in the total evaluation process. These MCQ tests are conducted with the help of Moodle learning management system.

iii. Programming Challenges

The programming challenges idea is inspired from Pratibha S. 2020 [13]. In programming challenge activity students need to implement the given programming statements in stipulated time duration. The problem statements are chosen carefully to ensure that they are completely unknown to the students. We conducted 2 programming challenge activities each of 20 marks. The weight assigned to this activity is 20% in the total evaluation process.

iv. Debugging Test

In debugging tests a code with bug (logical errors) in assigned to the student. The students are asked to identify the bug and remove it within stipulated time duration. Here students need to use debugging tools to identify and remove such bugs. We conducted 2 debugging tests each of 10 marks. The weight assigned to this activity is 10% in the total evaluation process.

Above four assessment methods are comprehensive and addresses different COs (Course Outcomes).

B. Course Results

After applying above assessment methods, final results of this Course were as follows. Final grade distribution of this course for two years (viz. 2019 and 2020) is as shown in figure 4. Total 72 students were evaluated in 2020, out of which 5 students got top grade (AA) whereas 6 students failed in this course. Overall grade distribution pattern follows the normal distribution curve; which is desirable. In 2020 the results are slightly improved with more number of students in top grades viz. 'AA' and 'AB'.

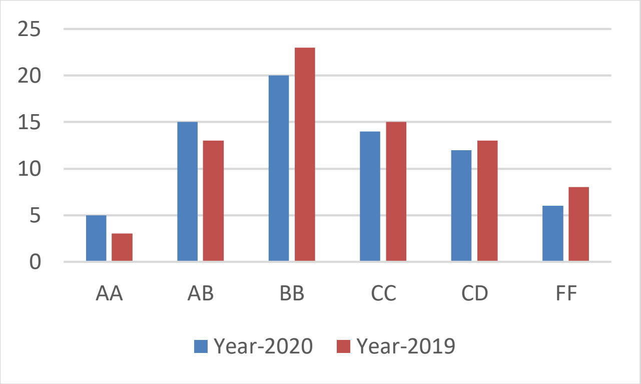


Figure 4 : Final Grade Distribution

C. Course Learning Outcome Attainments

Different assessment methodologies discussed earlier were mapped to different CO's and performance of all students in the same was noted. CLO attainment calculated using absolute method. The course learning outcome attainments for last two years is shown in figure 5. The attainment for all different CO's is increased in year 2020. There is significant increase in attainment of CO5 and CO1. This was due to focused efforts to improve student's awareness about computer systems and emphasis on debugging techniques and tools.

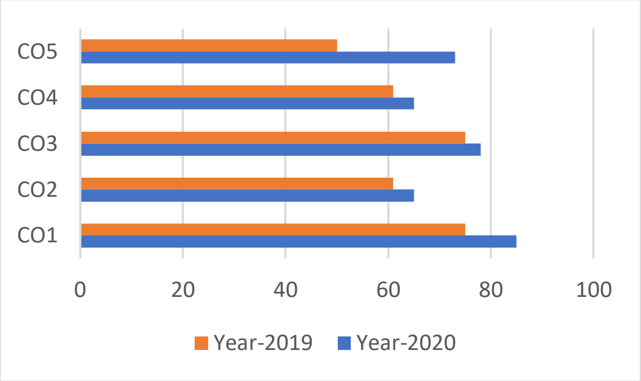


Figure 5 : Course Learning Outcome Attainments

6. Conclusion

Computer programming is an important course in first year engineering curriculum. Well planned and focused efforts are needed to ensure proper academic delivery of this course. Making students friendly and well aware about computer systems is an essential pre-requisite for computer programming course. This work discusses method to address this requirement systematically. Logic building, problem solving and debugging skills need to be developed among students at the end of this course. This work discusses different approaches which are useful to develop these skills. It is observed that academic results and overall course learning outcome attainment are improved after applying the proposed approach.

References

1. P. Chenna Reddy (2015) Analysis of Teaching Computer Programming in Indian Context, Journal of Engineering Education Transformations, Volume 28, No.4, April 2015.
2. https://www.aicte-india.org/education/model-syllabus
3. Abhiram Ranade (2016) An Approach to Introductory Programming, Journal of Engineering Education Transformations, Volume 29, No.3, January 2016.
4. Sunitha. P, Uma Boregowda, A.Geethakiran, Margaret. R.E, Vinod. A. M, Effective Teaching through Programming Assignments, Journal of Engineering Education Transformations, Volume, No, Month 2015
5. Uma Boregowda, Sunitha P, A Geetha Kiran, A Practical Approach to Make Computer Laboratory Courses more Effective and Interesting through Student Mentoring, Journal of Engineering Education Transformations, Volume 31, No. 1, July 2017.
6. Varsha T. Lokare, Prakash M. Jadhav, Snehal S. Patil, An Integrated approach for teaching Object Oriented Programming (C++) course, Journal of Engineering Education Transformations, Volume 31, No. 3, January 2018.
7. S. Kannan, D. Sumathi, T. Prabakaran, A Study on Challenges and Opportunities in Teaching Programming Subject to First Year Computer Science and Engineering Students: In the perspective of faculty and student, Journal of Engineering Education Transformations, Volume 31, No. 3, January 2018.
8. Jayashree A. Patil, Shubhangi P. Kale, Impact of ICT Tools in Logic Development of Computer Programming Skills Blooms Taxonomy, Journal of Engineering Education Transformations, Volume 33, No.1, July 2019
9. https://en.wikipedia.org/wiki/Bloom%27s_taxonomy
10. Dr. Paresh Tanna, Dr. Nirav Bhatt, Sweta Patel, An Innovative Approach for Learning and Evaluating Programming-Oriented Courses, Journal of Engineering Education Transformations, Volume 33, No. 3, January 2020.
11. Brian W. Kernighan, Dennis Ritchie, C Programming Language, Person Education, 2nd Edition.
12. E. Balagurusamy, Programming in ANSI C, Tata McGraw-Hill Education Published Date: 2008.
13. Pratibha S. Yalagi, Rashmi K. Dixit, Manisha A. Nirgude, Enhanced Programming Learning Model (EPLM) through Continuous Collaborative Coding (CCC) Practice, Journal of Engineering Education Transformations, Volume 33, January 2020.