# PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming

**Chapter** · January 2019

DOI: 10.1007/978-3-030-01659-3_34

**3 authors:**

Sohail Iqbal Malik
Al Buraimi University College
**77** PUBLICATIONS **805** CITATIONS
SEE PROFILE

Roy Mathew
Al Buraimi University College
**30** PUBLICATIONS **149** CITATIONS
SEE PROFILE

Maytham Hammood
Tikrit University
**19** PUBLICATIONS **127** CITATIONS
SEE PROFILE

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

# PROBSOL: A Web-based Application to Develop Problem-solving Skills in Introductory Programming

Sohail I. Malik[1], Roy Mathew[1] and Maytham M. Hammood[1,2]

[1]Buraimi University College, Al-Buraimi, OMAN
[2]Computer Science Dept., Tikrit University, IRAQ

{Sohail,Roy,Maytham}@buc.edu.om

**Abstract.** Problem-solving skills are considered as an important part of learning to program for novice programmers. Different techniques such as pseudocode and flowchart are used to develop these skills. In this study, a web-based application ‘PROBSOL’was developed and offered to the novices in an introductory programming course to acquire problem-solving skills. The PROBSOL application is based on pseudo-code technique. One of the advantages of this application for novices is to focus on problem domain without worrying about the syntax of the programming language. Practice questions related to all the topics included in the introductory programming course were prepared and offered in the application. The results show a positive impact on the students’ learning outcomes and attrition rate. Students (treatment group) who used the PROBSOL application show better results compared to the control group.

**Keywords:**Problem-solving skills, Pseudo code, Introductory programming.

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

## 1 Introduction

Programming is an important and fundamental skillthat needs to be developed to meet the requirements of an information society. Programmers are considered to be builders of the information society. They develop software applications consumers' use every day, multimedia applications which provide constant streaming and entertainment information, and complex systems which perform essential tasks.

The Bureau of Labor Statistics [1] predicts high demand for programmers in the employment market in coming years. To meet the requirements of the market, computer science (and related disciplines) degree programs emphasise the programming field by including programming studies in the first semester of the degree programs. To become a programmer requires the development of multiple skills including critical thinking, problem-solving, beingdetailed-oriented, as well as learning the syntax and semantics of the programming language. For novices, to acquire all these skills is considered a challenging and difficult task. They have to focus on learning bothproblem-solving strategies and the syntax and semantics of the programming language to acquire these skills.

High failure and drop-out rates from introductory programming (IP) courses continue to beofsignificant concern to computer science disciplines despite extensive research attempting to address the issue[2][3][4][5][6][7]. Researchers have identified that novice programmers spend more time trying to understand programming language syntax and semantics (programming knowledge) [8] rather than paying attention to planning the problem (problem-solving strategies) [8][9][10][11][12]. Koulouriet [13] argued that 'teaching problem solving before programming yielded significant improvements in student performance' (p. 1).

In this study, a web-based application 'PROBSOL' was developed and offered to the novices in an introductory programming course to acquire problem-solving skills. The PROBSOL application is based on pseudo-code technique. One of the advantages of this application for novices is to focus on problem domain without worrying about the syntax of the programming language.

This paper is divided into a number of sections.It starts with a review of theliterature, followed by a brief introduction to the PROBSOL application. The methodology used for this study is then described, and research results are reported and discussed. The paper concludes with a summary of the outcomes.

## 2 Literature Review

De Raadt [8] defines problem-solving as 'A mechanism for achieving a solution to a programming problem' (p. ix). This definition is also applicable in our context. It is

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

commonly accepted that teaching problem-solving techniques is not an easy task [14]. Additionally, different researchers have shown that students in IP courses donotconcentrate on developing problem-solving skills and instead spend more time on coding the programming problems [15].

The ACM-IEEE Joint Task Force on Computing Curricula [16] provides guidelines for computing undergraduate programs. The latest revision, Computer Science Curricula2013 (CS2013), provides comprehensive guidance on curricular structure and development in a variety of educational contexts [16].The CS2013 body of knowledge is organised into 18 knowledge areas (KA). The most important programming related KAs include Programming Languages (PL), Software Development Fundamentals (SDF), Algorithms and Complexity (AL) and Software Engineering (SE). Computer programming concepts and skills topics are introduced in these KAs for introductory and advanced computer science programming courses.Software Development Fundamentals (SDF) is a newly introduced KA in CS2013. It focuses on the entire software development process in introductory programming. Due to its broad spectrum, it includes KAs which could be included in other software-oriented KAs (e.g. programming constructs and problem-solving (PL), development methodologies (SE), and algorithm analysis (AL)). SDF provides basic concepts, and those KAs cover advanced topics in the above-mentioned areas.Computer programming knowledge (syntax and semantics) and problem-solving strategies are given equal attention in these KAs. Problem-solving skills are also included as one of the characteristics of computer science graduates which means that graduates should know how to identify and design solutions to real world problems instead of just writing code [16]. The Australian Computer Society [17] also suggested emphasising both problem-solving strategies and programming knowledge in the teaching and learning of programming languages.

De Raadt [18] reviewed forty textbooks prescribed to students in introductoryprogramming.Problem-solving topics were covered in varying degrees in the analysed books. De Raadt[18] mention that only six books out of forty investigated integrated problem-solving aspects throughout, implying that most of the analysed books focused only on language syntax. Some authors provide a large number of examples and put less emphasis on problem-solving instructions. Some books cover problem-solving topics in early chapters, but they are not integrated into the remainder of the book. There are a small number of books where problem-solving is integrated throughout the books using case studies and examples [18].

Winslow [19] discussed problem-solving learning processes for students. He emphasised that, for novices, understanding programming language syntax, semantics and problem-solving skills are equally important. He divided program problem solving into four steps:

1) ‘Understand the problem
2) Determine how to solve the problem:

a) in some form, and

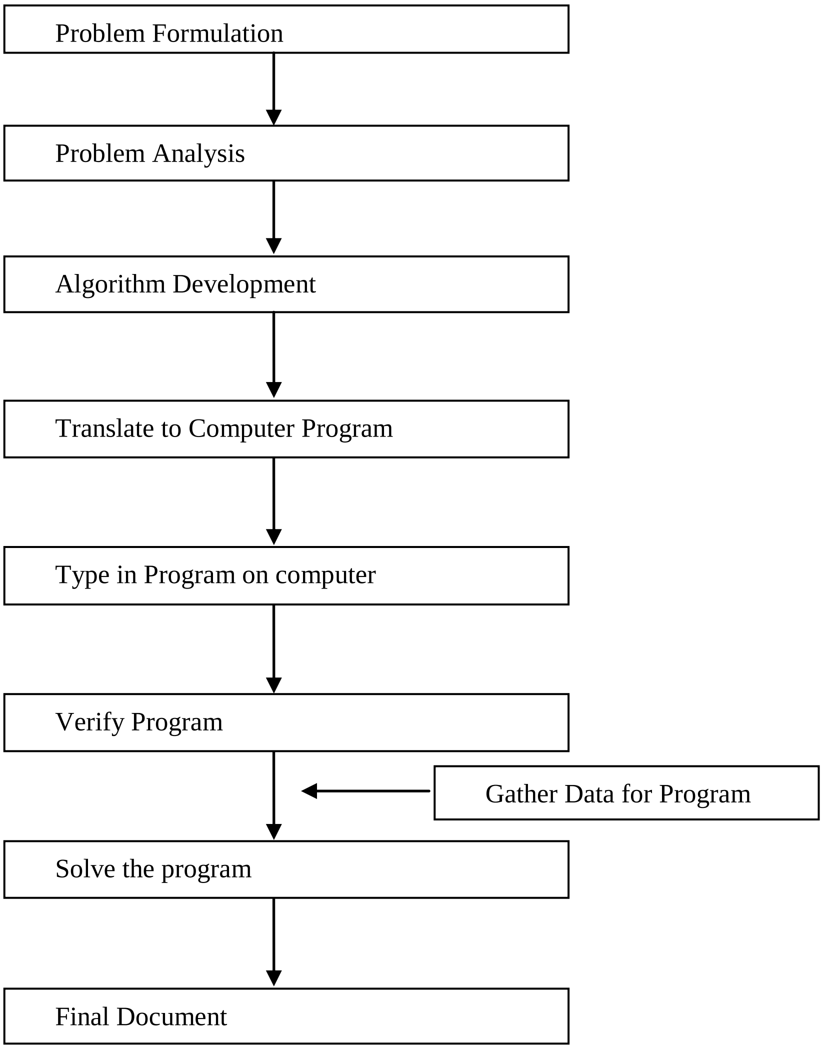
b) in computer compatible formNote that novices have trouble going from ‘a’ to ‘b’)

3) Translate the solution into computer language program, and

4) Test and debug the program’ (p. 19)

Soloway [20] suggested *Goals* and *Plans* as a problem-solving strategy for analysing problems and constructing programs. ‘With a given problem, the process begins with the instructor determining the goals that need to be achieved to solve the problem. These goals are then mapped to *plan’*(p. 26)[8].

Hyde [21] developed the Problem Solving Process (PSP) for an introductory programming course which promoted problem-solving skills in students. PSP consists of the steps shown in Figure 1. It is clear that PSP requires students to finish four steps before typing the program on thecomputer. The PSP promotes problem-solving skills in students.



Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

**Fig.1**Problem Solving Process (PSP) Hyde [21]

Koulouriet [13] conducted a study in an introductory programming course to determine the benefits of problem-solving training before programming. They compared the performance of two groups of students with and without problem-solving training. They revealed that students provided with problem-solving training performed better than those without problem-solving training. They discussed a possible explanation was that novice programmers start working with syntax and semantics of the language without considering oranalysing the problem statement. The problem-solving training helps students to understand and interpret the problem in terms of devising a programming solution, which they can then translate into lines of code. This argument is also supported by [22].

Different tools have been developed to teach the problem-solving skills to the novices in an introductory programming course. Malik [23] developed the ADRI editor based on the four stages of the ADRI approach and offered in an introductory programming course. The editor offers pseudo-code and flowchart techniques to develop the problem-solving skills. Novices have to write the pseudo-code for a given problem statement. The editor does not support any feature to identify errors or provide thecorrect solution for the given problem statement. Oda [24] proposed a method to automatically generate a pseudo-code from a source code by using the statistical machine translation (SMT) framework. Taheri [25] discussed that progranimate is a web-based environment which uses flowcharts to find solutions for basic programming problems for novices.

## 3 Research Question

The study focuses on determining the impact of the PROBSOL application on students learningprocess in the introductory programming course.

The research questionbeing addressed in this study is:

*RQ: What is the impact of the PROBSOL application on the students learning outcomes in the introductory programming course?*

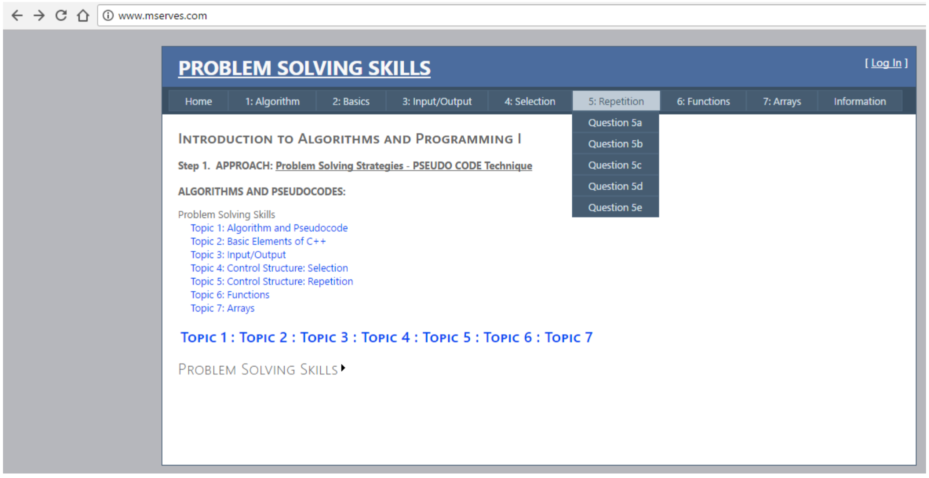
The researchquestion is explored in section 6.

## 4 Introduction to the PROBSOL application

The PROBSOL is a web-based application based on the pseudo-code technique. The application can be used to develop problem-solving skills to the novice programmers

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

in introductory programming. Figure 2 shows the main page of the PROBSOL application.



INTRODUCTION TO ALGORITHMS AND PROGRAMMING I

Step 1. APPROACH: Problem Solving Strategies - PSEUDO CODE Technique

ALGORITHMS AND PSEUDOCODES:

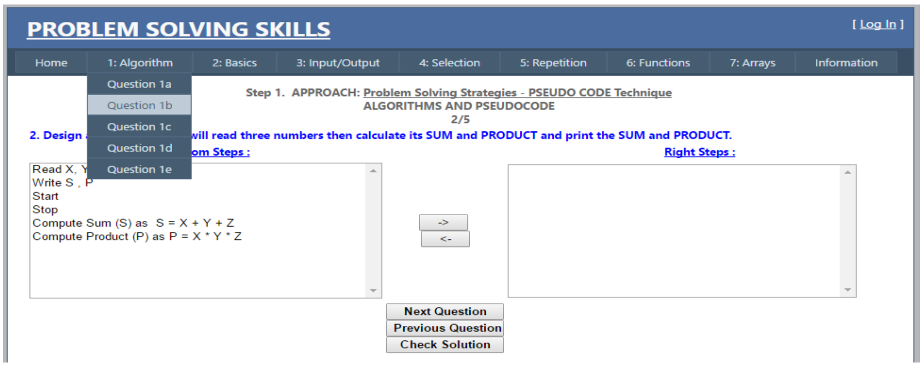
Problem Solving Skills
- Topic 1: Algorithm and Pseudocode
- Topic 2: Basic Elements of C++
- Topic 3: Input/Output
- Topic 4: Control Structure: Selection
- Topic 5: Control Structure: Repetition
- Topic 6: Functions
- Topic 7: Arrays

TOPIC 1 : TOPIC 2 : TOPIC 3 : TOPIC 4 : TOPIC 5 : TOPIC 6 : TOPIC 7

PROBLEM SOLVING SKILLS

**Fig.**2.Main page of the PROBSOL application

Practice questions related to all the topics included in the introductory programming course were prepared and offered in the application. Figure 3 depicts when a user selects a question related to topic 1 in the IP course. A problem statement is shown for each question.



Step 1. APPROACH: Problem Solving Strategies - PSEUDO CODE Technique
ALGORITHMS AND PSEUDOCODE
2/5

2. Design will read three numbers then calculate its SUM and PRODUCT and print the SUM and PRODUCT.

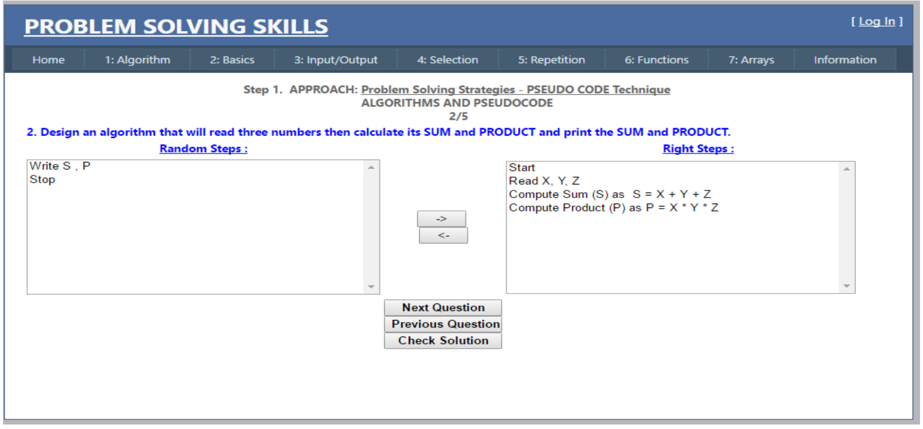
om Steps :

Right Steps :

```
Read X, Y
Write S , P
Start
Stop
Compute Sum (S) as  S = X + Y + Z
Compute Product (P) as P = X * Y * Z
```

**Fig.3.**The PROBSOL application with selected question

The application uses three list boxes (Random steps, Right Steps and Solution). The Random steps list box shows all the pseudo-code steps solution for a given problem statement in a random manner. The user selects the steps in the Random steps list box one by one and moves it to the Right Steps list box by clicking Right arrow button as shown in figure 4. The solution list box (depicted in figure 5) shows the correct pseudo-code solution for a given problem statement.



Step 1. APPROACH: Problem Solving Strategies - PSEUDO CODE Technique
ALGORITHMS AND PSEUDOCODE
2/5

2. Design an algorithm that will read three numbers then calculate its SUM and PRODUCT and print the SUM and PRODUCT.

Random Steps :

```
Write S , P
Stop
```

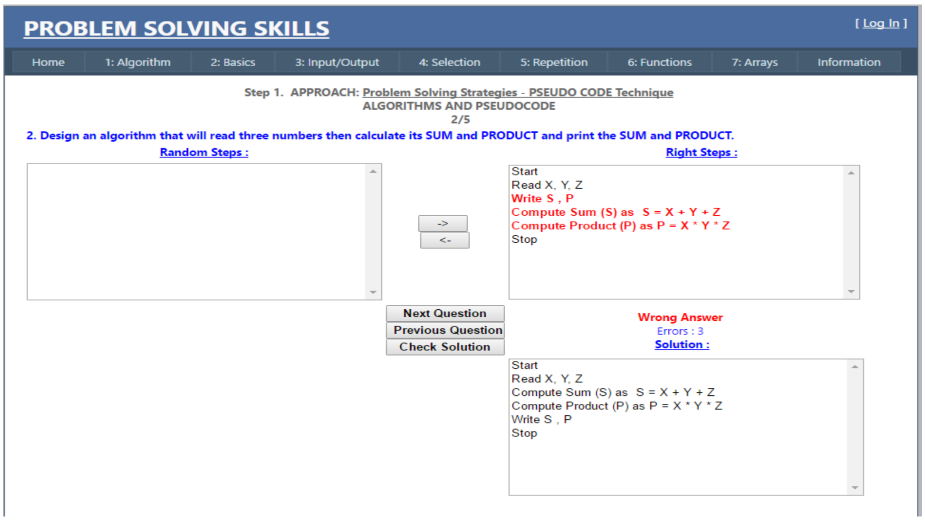
Right Steps :

```
Start
Read X, Y, Z
Compute Sum (S) as  S = X + Y + Z
Compute Product (P) as P = X * Y * Z
```

**Fig.4.**The PROBSOL application with intermediated solution

When the user clicks the ‘Check Solution Button’, and the answer is incorrect, the lines or steps containing errors are displayed in Right Steps list box with red colour, indicating that those lines are there with mistakes(depicted in figure 5). Those lines of codes with red colourare not placed in its correct position, and the order is wrong in Right Steps list box. The Solution list box shows the correct solution of the question with its proper intended order. It also evaluates your solution, and the number of lines with mistakes are shown with its calculated number of Errors (e.g., Error No. 3) under the Wrong Answer part which is displayed between Right Steps and Solution list boxes.

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34



Step 1. APPROACH: Problem Solving Strategies - PSEUDO CODE Technique
ALGORITHMS AND PSEUDOCODE
2/5

2. Design an algorithm that will read three numbers then calculate its SUM and PRODUCT and print the SUM and PRODUCT.

Random Steps :

Right Steps :

```
Start
Read X, Y, Z
Write S , P
Compute Sum (S) as  S = X + Y + Z
Compute Product (P) as P = X * Y * Z
Stop
```

Wrong Answer
Errors : 3
Solution :

```
Start
Read X, Y, Z
Compute Sum (S) as  S = X + Y + Z
Compute Product (P) as P = X * Y * Z
Write S , P
Stop
```

**Fig.5.**The PROBSOL application with three list boxes

Different warning and error messages are displayed in the application for example when the user clicks Right Arrow Button without selecting any option from the Random Step list box, or the user clicks Right Arrow Button when the Random Step list box is empty.

## 5 Methodology

The research question was exploredby comparing the final grades of students over two semesters to determine the impact of the PROBSOL application on the students' achievements. The application was offered to the students of the IP course in the semester 2, 2016-17. Students' grades were compared with the previous semester (semester 1, 2016-17) to determine the impact of the application if any of the students' achievements. The grades were compared based on the criteria described in table 1. Students were divided into three categories, high, medium and low achievers based on their marks. More information about the three categories is provided in section 6.

Data collectionwasperformed after obtaining ethical approval from BuraimiUniversity College.Students' grades were collected after the de-identification process performed by the Registration department at Buraimi University College.

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

## 6 Results and Discussion

The grades of students who finished the IP course after introducing the PROBSOL application were compared with the previous semester grades. The purpose is to determine the impact of the application on the students' achievements.

Students' grades were analysedfollowing the three categories proposed by [26]. Table 1 depicts the three categories.

**Table 1.**Three Categories of students (adapted from [26])

| Category | Mark Range |
|---|---|
| High Achiever | 85-100 |
| Medium achiever | 65-84 |
| Low achiever | 50-64 |

Malik et al. [26] allocated students into three categories (high, medium and low achievers) based on their marks. Students who obtain marks in between 50 to 64 are considered aslowachievers; those in between 65 to 84 are medium achievers, where students in between 85 to 100 are considered as high achievers.

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

Figure 6 shows the comparison of students' grades in the IP course.

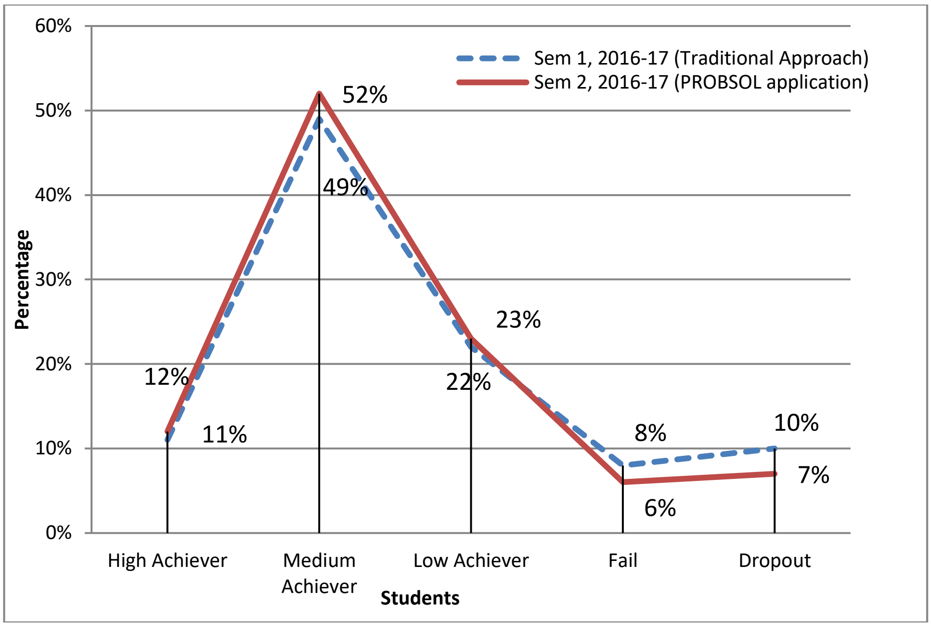


**Fig.6.** Impact of the PROBSOL application on students' achevements

The PROBSOL application provides a small, but positive improvement in the highachiever category, with 12% of the students who passed the course with thePROBSOL application in this category compared with 11% for the traditional approach. The medium achiever category shows an increase of 3%; 49% of the students reached the medium achiever category compared with 52% for the PROBSOL application. There is a smallpositive improvement in the low achiever category of 1%; 23% of students with the PROBSOL applicationcompared with 22% for the traditional approach. The failure rate with the PROBSOL application semester is 6% compared with the 8% for the traditional approach semester, a reduction of 2%. Likewise, the dropout rate (those students who do not complete the course) with the PROBSOL applicationsemesteris7% compared with the 10% for the traditional approach semester, and it is reduced by 3%. This result is consistent with previous findings of [27].

Overall, the PROBSOL application provides a positive impact on the teaching and learning process of the IP course. It emphasises students to avoid programming shortcut (ProblemStatement →Codes) and follows proper programming process (Problem Statement →Solution plans→Codes). It also provides an opportunity to the students to focus only on the solution for the given problem statement without much worry about the syntax of the programming language. Moreover, it emphasizes students to pay equal attention to the problem solving strategies besides the syntax and semantics of the programming language. This argument is supported by [28][29][30][31].

The design of the PROBSOL application is simple and user-friendly which also facilitates the learning process. Students only need to select appropriate steps in the given solution and click the appropriate button to move the step from one list box to another. This simple process helps the students to focus more on the solution of the given problem statement instead of commands. Moreover, the PROBSOL application is web-based which also facilitates student attitudes towards learning. Paramasivam [32] concluded that computers in instruction process improve student attitude towards learning. Moreover, female students in computer science department perceived web as the easiest area of study [33]. Eldow [34] also suggested to use E-learning methods in the teaching process.

Practice questions related to all the topics in the IP course are embedded in the application which facilitates the learning process. The questions are categorised based on the teaching topics of the course. Students can choose questions by topic.

The result demonstrates that the PROBSOL application engages students in understanding the programming concepts. Students performed better in all the high, medium and low achievers categories. The application also promotes deep-learning among students while solving the given problem statement. Moreover, the application provides themore detail-oriented experience of the programming domain.

The failure rate was reduced for the semester offered with the PROBSOL application compared to the semester offered with the traditional approach. The application develops students' interest in the course by providing them a clear picture of the programming domain. Ultimately, this trend will help students' enrolment and retention in the computer science field. This result is consistent with previous findings of [31].

The dropout rate was reduced also for the semester offered with the PROBSOL application compared with the previous semester offered without the PROBSOL application. The PROBSOL application helps students to develop their programming logic from the beginning of the semester which ultimately develop their confidence and interest in the learning process.

## 7 Conclusion

The PROBSOL application was introduced in the introductory programming course for students to acquire problem-solving skills. The application is based on pseudocode technique. The application is web-based, easy to access and use. The application includes a number of questions related to all the topics in the course.

The PROBSOL application provides a positive impact on the teaching and learning process of the IP course. The students performed better in the high, medium and low achievers categories. Moreover, the attrition rates (failure and dropout) were improved in the semester which was offered with the PROBSOL application compared with the previous semester which was offered with the traditional approach.

The PROBSOL applicationengages students in understanding the programming concepts. It promotes deep-learning and discourages programming shortcut (ProblemStatement→Codes). It also encourages students to follow proper programming process (Problem Statement →Solution plans→Codes).

## References

1. Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2014-15 Edition, Computer Programmers, http://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm, last Accessed 28/8/2015.
2. Guzdial, M., Soloway, E.: Log on education: teaching the Nintendo generation to program.Communications of the ACM, 45(4), 17-21 (2002)
3. Lahtinen, E., Ala-Mutka, K., Järvinen, H.M.: A study of the difficulties of novice programmers. ACM SIGCSE Bulletin, 37(3), 14-18. (2005)
4. Sykes, E.R.: Determining the Effectiveness of the 3D Alice Programming Environment at the Computer Science I Level. Journal of Educational Computing Research, 36(2), 223-244 (2007)
5. Yadin, A.: Reducing the dropout rates in an introductory programming course. ACM Inroads, 2(4), 71-76 (2011)
6. Watson, C., Li, F.W.B.: Failure rates in introductory programming revisited, In Proceedings of ITiCSE, ACM, Uppsala, Sweden (2014)
7. Zingaro, D.: Examing interest and grades in computer science 1: A study of pedagogy and achievement goals, Transactions of Computer Education, 15(3), ACM (2015)
8. De Raadt, M.: Teaching programming strategies explicitly to novice programmers', PhD thesis, University of Southern Queensland, Australia, retrieved June 2013, USQ ePrints (2008)
9. Webster, M.: Overview of programming and problem solving, Merriam-Webster's Collegiate Dictionary, Tenth Edition, retrieved on15/7/ 2013, http://computerscience.jbpub.com/vbnet/pdfs/mcmillan01.pdf (1994)
10. Kölling, M., Rosenberg, J.: BlueJ - A Language for Teaching Object-Oriented Programming, In Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education, pp. 190-194, St. Louis, MO, USA (1996).
11. Ala-Mutka, K.: Problems in learning and teaching programming – a literature study for developing visualizations in the codewitz-minerva project, Codewitz needs analysis, retrieved 20/12/2013, http://www.cs.tut.fi/~edge/literature_study.pdf (2004)
12. Iqbal, S., Harsh, O.k.: A self review and external review model for teaching and assessing novice programmers. International Journal of Information and Education Technology, 3(2), 120-123 (2013)

13. Koulouri, T., Lauria, S., Macredie, R.D.: Teaching introductory programming: A Quantitative Evaluation of Different Approaches. Transactions of Computer Education, 14(4), ACM (2015)
14. Allan, V.H., Kolesar, M.V.: Teaching computer science: a problem solving approach that works, SIGCUE outlook, ACM, 25(1-2), 2-10 (1997)
15. Yoo, J., Yoo, S., Seo, S., Dong, Z., Pettey, C.: Can we teach algorithm development skills?', In Proceedings of the 50th Annual Southeast Regional Conference, pp. 101-105, ACM, New York, (2012)
16. ACM-IEEE Joint Task Force on Computing Curricula,: Computer Science Curricula 2013, ACM Press and IEEE Computer Society press DOI: http://dx.doi.org/10.1145/2534860 (2013)
17. Australian Computer Society,: The ICT profession body of knowledge, retrieved 25/9/2015, <acs.org.au/__data/assets/pdf_file/0013/24502/The-ICT-Profession-Body-of-Knowledge-23-Sept-2013.pdf (2013)
18. De Raadt, M., Toleman, M., Watson, R.: Textbooks Under Inspection, University of Southern Queensland, Australia, retrieved May 2013, http://eprints.usq.edu.au/167/1/TechReport_Draft_10.pdf (2005)
19. Winslow, L.E.: Programming pedagogy—a psychological overview, ACM SIGCSE Bulletin, 28(3), 17-22 (1996)
20. Soloway, E.: Learning to Program = Learning to construct mechanisms and explanations, Communications of the ACM, 29(9), 850-858 (1986)
21. Hyde, D.C., Gay, B.D., Utter, D.: The integration of a problem solving process in the first course', In Proceedings of the tenth SIGCSE technical symposium on Computer science education, pp. 54-59, ACM, (1979)
22. Malik, I.S., Coldwell-Neilson, Jo.: A model for teaching an introductory programming course using ADRI, Education and Information Technologies, Springer, DOI: 10.1007/s10639-016-9474-0 (2017)
23. Malik, I.S.: Enhancing practice and achievement in introductory programming using an ADRI editor, In Proceedings of the IEEE International Conference on Teaching, Assessment and Learning for Engineering, pp. 32-39, IEEE, Thailand, 7-9 December, DOI: 10.1109/TALE.2016.7851766(2016)
24. Oda, Y., Fudaba, H., Neubig, G., Hata, H., Sakti, S., Toda, T., Nakamura, S.: Learning to Generate Pseudo-code from Source Code using Statistical Machine Translation, In Proceedings of 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), DOI: 10.1109/ASE.2015.36 , USA (2015)
25. Taheri, S.M., Sasaki, M., Ngetha, H.N.: Evaluating the Effectiveness of Problem Solving Techniques and Tools in Programming, In Proceedings of Science and Information Conference (SAI) IEEE, DOI: 10.1109/SAI.2015.7237253 ,UK (2015)
26. Malik, I.S., Coldwell-Neilson, Jo.: Impact of a new teaching and learning approach in an introductory programming course, Journal of Educational Computing Research, SAGE, 55(6), 789-819, DOI: 10.1177/0735633116685852 (2017)
27. Malik, I.S., Coldwell-Neilson, Jo.: Comparison of traditional and ADRI based teaching approaches in an introductory programming course. Journal of Information Technology Education: Research, 16, 267-283. http://www.informingscience.org/Publications/3793 (2017)

Malik S.I., Mathew R., Hammood M.M. (2019) PROBSOL: A Web-Based Application to Develop Problem-Solving Skills in Introductory Programming. In: Al-Masri A., Curran K. (eds) Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development). Springer, pp 295-302, https://doi.org/10.1007/978-3-030-01659-3_34

28. Malik, I.S.: Improvements in Introductory Programming Course: Action Research Insights and Outcomes, Systemic Practice and Action Research, Springer, DOI: 10.1007/s11213-018-9446-y (2018)
29. Malik, I.S., Mathew, R., Chowdhury, M.U., Coldwell-Neilson, Jo.: Impact of Assurance of Learning (AOL) in programming course for novices, Proceedings of 2nd BCS The Chartered Institute for IT International Conference, BCS Learning and Development, United Arab Emirates (2014)
30. Malik, I.S., Coldwell-Neilson, Jo.: Gender differences in an introductory programming course: New teaching approach, students' learning outcomes, and perceptions, Education and Information Technologies, Springer, DOI: 10.1007/s10639-018-9725-3 (2018)
31. Malik, I.S.: Role of ADRI model in teaching and assessing novice programmers, PhD thesis, Deakin University, http://hdl.handle.net/10536/DRO/DU:30088862 (2016)
32. Paramasivam, G., Manohar, J., Malik, I.S.: Bringing Technology into the classroom: Receptivity and Satisfaction level of faculty members Using Smart board – An Experimental Study, In Proceedings of International Conference on Applied Information and Communication Technology, Oman (2014)
33. Malik, I.S., Al-Emran, M.: Social Factors Influence on Career Choices for Female Computer Science Students, International Journal of Emerging Technologies in Learning, 13 (5), pp. 56-70, 2018
34. Eldow, A., Shakir, M., Abdullah, M.Y., Iqbal, S.: Using Information Communication Technology as a Teaching tool in Sudanese Governmental Universities of Khartoum State, International Journal on Informatics Visualization, 1(4), pp.150-156