Integrating Class and Laboratory with Hands-on Programming: Its Benefits and Challenges

Vidya Handur[1], Priyadarshini D Kalwad[2], Mahesh S Patil[3], Vishwanath G Garagad[4], Nagaratna Yeligar[5], Praveenraj Pattar[6], Deepak Mehta[7], Preethi Baligar[8], G.H Joshi[9]
Department of Computer Science and Engineering
B.V.B.College of Engineering and Technology
Hubballi-580 031, Karnataka, INDIA.
[1]vidya_handur@bvb.edu, [2]priyadarshini@bvb.edu, [3]mahesh_patil@bvb.edu, [4]vishwanath_gg@bvb.edu, [5]nagaratna@bvb.edu, [6]praveenraj@bvb.edu, [7]deepak.mehta@bvb.edu, [8]preethi.b@bvb.edu, [9]ghjoshi@bvb.edu

***Abstract*—The traditional approach of teaching programming courses is teachers centric where students are passive learners. Also for such courses, the laboratory and classes are conducted separately. This paper focuses on integrating classroom and laboratory with hands-on for programming course. This approach is student centric which brings in active learning. However it has been less researched area and adequate literature is not available on 'hands-on science' specifically for programming course. The work presented in this paper also extends from our previous work where debugging, code optimization and testing are emphasized. In the proposed approach each teaching session consisted of hands-on teaching and activities to support it. This requires strategic planning of the course and its delivery as discussed in the paper. Applying this approach, there is increase in the number of students scoring higher grades and overall improvement in the average. The paper also discusses the benefits and challenges of hands-on discovered in this case study that triggers further research.**

***Index Terms*— Hands-on, active learning, students centric, course design, integrating laboratory and classroom, programming course.**

I. Introduction

The present scenario shows that teaching is continued with the traditional approach. i.e delivery of lecture in large classrooms. The approach includes chalk-and-talk or pen-and-paper teaching [1] [2]. There is need for transformation in such teaching methods to emphasize relevance and problem-solving approach of engineering.

Currently most of the programming courses are taught with separate class and laboratories or without laboratories where students are passive learners. Some attempts have been made where programming courses have integrated class and laboratories with hands-on teaching that incorporates active learning. However this has been less researched area and adequate literature is not available on 'hands-on science' specifically for programming course. Such Hands-on teaching for a programming course has many benefits as well as challenges that are presented in this paper. Thus designing programming course with hands-on requires thorough understanding of 'hands-on science' and its limitation (Given in section 2).

This paper presents a case study where class and laboratories are integrated with hands-on teaching for a programming course. The paper presents the course design that incorporates available 'hands-on science' literatures and highlights the benefits and challenges identified through this case study. The course is taught for the first year undergraduate students who have little or no programming background. The aims of this course are to develop problem-solving skills and programming skills among the students.

The section 2 describes the literature about 'hands-on science' and 'class-laboratory integrations' in general. The section 3 describes case study. i.e specific aspects of course design concerned to hands-on and class-laboratory integrations. Section 4 discusses results, hands-on benefits and challenges. Finally section 5 concludes with new research questions.

II. Literature Survey

In traditional classes for programming course, Students may understands concepts incorrectly which will lead into misinterpretations of the concepts [3] [4] i.e. correct transfer of knowledge from teachers to students cannot be guaranteed [5]. As per constructivist theory knowledge can be transmitted to students through engagement in building the knowledge as in [6] [7]. But in traditional classes where programming courses are taught with lectures and demonstration, the students are passive and transfer of knowledge cannot be guaranteed.

Programming courses must facilitate conceptual understanding in students. Conceptual understanding can be defined as ability to observe, interpret and summarize a concept. It is also possessing of knowledge and its understanding as shown in [8]. Building conceptual understanding among the students cannot be simply achieved by lectures and students memorizing the concepts. In order to achieve it, contextual and experiential learning environment should be provided to students as in [8]. One way of ensuring conceptual understanding in students can be integration of active learning methods like hands-on in to regular learning process [5].

A. Hands-on Science:

Hands-on is way of teaching where students have resources for testing what they learn. i.e. students learn by trying and manipulating the resources [9]. It is also defined as learning with materials where students modify, handle and test the

978-1-5090-1062-2/16 $31.00 © 2016 IEEE
DOI 10.1109/MITE.2016.47

learning's with materials [9]. This helps in developing muscles, perceptual and psychomotor skills [9]. Hands-on teaching brings lab tasks to classrooms or vice versa, where students are involved in using, changing and testing materials and observe the working modules[10].In traditional teaching teacher is the center while in hands-on , students are the center of focus. i.e. students are actively involved trying out the learning's and observing the things. This approach shall create interest in students learning and ignites thinking [10].

B. Benefits of hands-on Teaching

Following are the benefits of Hands-on teaching.

i. Hands-on keeps students engaged which helps in better learning and longer retentions [11].
ii. Students can experience the process completely [12].
iii. Hands-on provides constructivist approach to in teaching. i.e it enables students to build their knowledge by doing [13].
iv. Hand-on increases the performance of students [9].
v. Hands-on helps those students to gain more who are challenged economically or academically [9].
vi. Students develop better attitude towards the course with hands-on teaching even though it is conducted poorly [14].
vii. Hands-on is active learning which improves student's creativity and problem-solving skills. It also improves the communication skills [15]

In traditional teaching there is a student's engagement gap in the process of learning and such gap can be filled by hands-on teaching [16]. Hands-on incorporates 21st century learning skills like collaboration, communication, critical and creative thinking.

Hands-on teaching generates interest among students towards the course being taught [17]. Interest can be defined as students-materials relationship that is generated during the student's interaction with materials [18] [19]. This interest of students is important factor that can decide the quality of learning and its results in evaluations [20] [21]. Students also feel that hands-on sessions are more interesting than teacher's instructions, watching videos or listening to audios [22]. But Interest generated in hands-on can be varying. It mainly depends on the student's involvement which is based on his/her positive cognition and emotions [18] [23]. Therefore interest developed depends if students experience and learning is joyful. But the negative experience of student can also reduce the interest towards the course. Example: student unable to complete the tasks, de-motivation and emotional discomforts [19].

Hands-on can be more challenging for students, if it involves critical thinking and analysis [24]. Such hands-on sessions may create low confidence in students for those who are used to hands-on teaching. Each hands-on teaching is not one and the same in terms of student's involvement, learning achieved and the interest it generates. i.e. each hands-on requires different skills of students and planning by teachers [10]. So they vary in terms of time duration, teachers planning skills and social relationship among students [25].

C. Integration of laboratory and class-room

In computer science education many programming courses are taught in traditional way, where lectures and laboratory are conducted separately. Traditional classes do not help students in learning the concepts being taught. Also, Students fail to recognize the importance of the knowledge being transmitted. Conducting separate theory and lab classes creates gap between learning's and laboratory skills .i.e. students do not connect classroom learning's with laboratory skills [26]. Therefore class-room can be integrated in to laboratories for programming courses.

An integrated programming course can incorporate active learning's and hands-on programming. Learning in details with hands-on can be achieved through integrated laboratory classes. Such classes can also incorporate other type of learning's such as problem based learning (PBL), self-directed learning and e-learning. But such classes should not include video demonstration which can keep learners passive in the classrooms [26]. But it is also true that, not all classes can be integrated with laboratory. The paper [26] reveals that laboratory integration in class helps short-term and long-term retentions of students. This improvement is observed in short-term retention but performance in long-term retention is satisfactory.

III. Case Study

In traditional teaching students are passive. The programming concepts are taught in class-room and students find it difficult to apply the learning in the laboratories later. So there would be lack of connectivity between the learning and practicing. To overcome these problems, our novelty of pedagogy was to integrate class and laboratory with hands-on teaching.

The following are the course objectives framed for the novice approach:

i. Explain the process of problem solving using computers
ii. Ability to think logically with an algorithmic approach to solve specific problems.
iii. Develop operating familiarity with UNIX environment: Basic Unix commands, vi editor, C compiler.
iv. Write maintainable program using appropriate constructs of C programming language for a given problem.
v. Debug a given program for syntax and logical errors.
vi. Test the given program using the test plan.
vii. Review a given C program.
viii. Demonstrate code optimization skills.

A. Hands on Teaching for programming course

The approach of integrating classroom and laboratory with hands-on teaching in this case study has following features.

i. It consisted of two laboratory sessions per week each of 180 minutes duration. Total number of weeks being 13.
ii. In a session, there is one instructor demonstrating example with hands-on and three other instructors to help out students in coding.

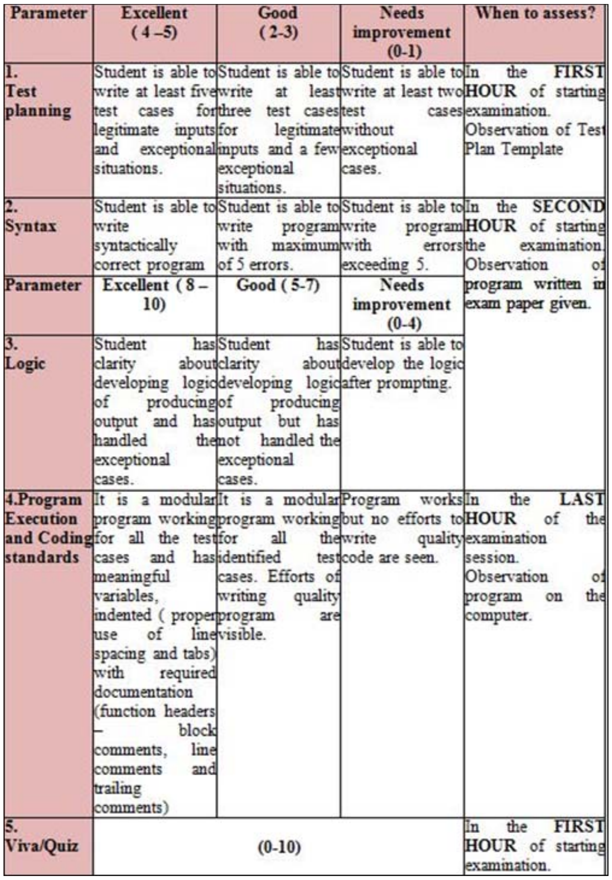
iii. Since the strength of the class was nearly 80, each instructor was assigned 20 students to mentor them in coding during the hands-on and evaluating them in assessments.

iv. Common content delivery materials are prepared across all the divisions.

v. More time allotted for hands-on than lecture. i.e a small lecture is given followed by hands-on for students to test the concepts. Then again small lecture on next concept with hands-on is given.

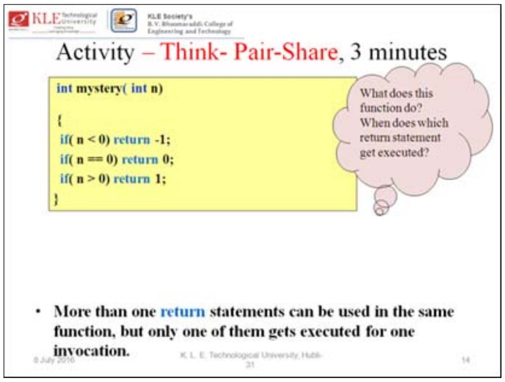
vi. The hands-on teaching also includes wake-up activities such as predicting the output/errors, think-pair-share, fill-up the blanks or small questionnaires. Examples are shown in fig 2 to fig 6. This helps students to reflect on their understanding.

In this hands on teaching, the students test what they learn. Hence this approach generates interest among students towards the course being undertaken. However spending entire time on hands-on is not good idea. Therefore the activities are conducted to help students to reflect on their understanding. This also gives instant feedback to students based on which instructor can adjust the content delivery.



Parameter	Excellent (4 –5)	Good (2-3)	Needs improvement (0-1)	When to assess?
1. Test planning	Student is able to write at least five test cases for legitimate inputs and exceptional situations.	Student is able to write at least three test cases for legitimate inputs and a few exceptional situations.	Student is able to write at least two test cases without exceptional cases.	In the FIRST HOUR of starting examination. Observation of Test Plan Template
2. Syntax	Student is able to write syntactically correct program	Student is able to write program with maximum of 5 errors.	Student is able to write program with errors exceeding 5.	In the SECOND HOUR of starting the examination. Observation of program written in exam paper given.
Parameter	Excellent (8 – 10)	Good (5-7)	Needs improvement (0-4)	
3. Logic	Student has clarity about developing logic of producing output and has handled the exceptional cases.	Student has clarity about developing logic of producing output but has not handled the exceptional cases.	Student is able to develop the logic after prompting.	
4.Program Execution and Coding standards	It is a modular program working for all the test cases and has meaningful variables, indented (proper use of line spacing and tabs) with required documentation (function headers – block comments, line comments and trailing comments)	It is a modular program working for all the identified test cases. Efforts of writing quality program are visible.	Program works but no efforts to write quality code are seen.	In the LAST HOUR of the examination session. Observation of program on the computer.
5. Viva/Quiz	(0-10)			In the FIRST HOUR of starting examination.

Fig 1. Rubrics Explanation for ESA 2015-16



KLE Technological University

KLE Society's B.V. Bhoomaraddi College of Engineering and Technology

Activity – Think- Pair-Share, 3 minutes

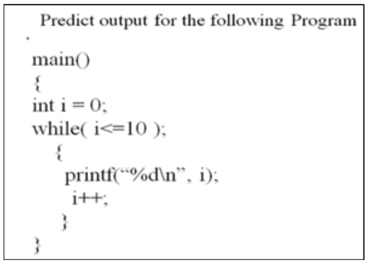
```
int mystery( int n)

{
 if( n < 0) return -1;
 if( n == 0) return 0;
 if( n > 0) return 1;
}
```

What does this function do? When does which return statement get executed?

- More than one return statements can be used in the same function, but only one of them gets executed for one invocation.

8 July 2016 K. L. E. Technological University, Hubli-31 14

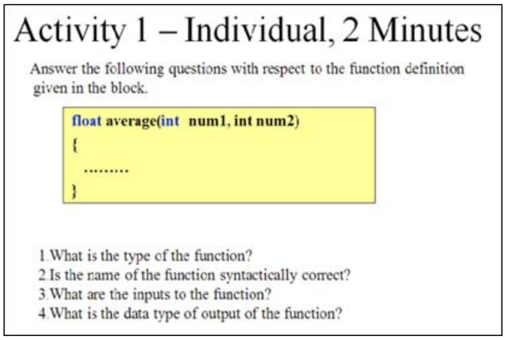
Fig 2. Snapshot of Activity conducted during session



Predict output for the following Program

```
main()
{
int i = 0;
while( i<=10 );
   {
     printf("%d\n", i);
       i++;
     }
}
```

Fig 3. Snapshot of the predicting the output during session



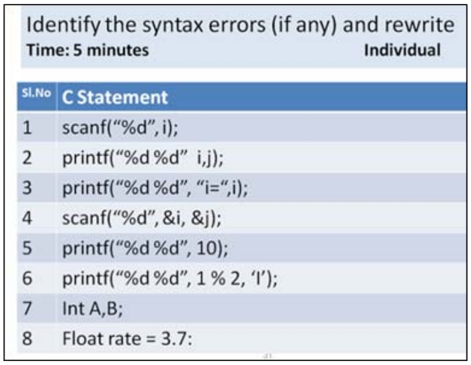
Activity 1 – Individual, 2 Minutes

Answer the following questions with respect to the function definition given in the block.

```
float average(int  num1, int num2)
{
  ………
}
```

1 What is the type of the function?
2 Is the name of the function syntactically correct?
3 What are the inputs to the function?
4 What is the data type of output of the function?

Fig 4. Snapshot of Activity conducted during session

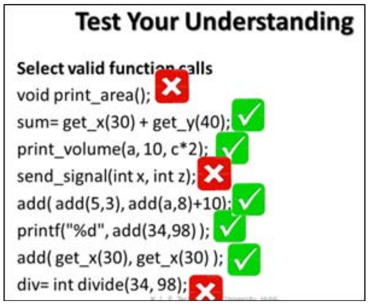


Identify the syntax errors (if any) and rewrite

Time: 5 minutes Individual

Sl.No	C Statement
1	scanf("%d", i);
2	printf("%d %d" i,j);
3	printf("%d %d", "i=",i);
4	scanf("%d", &i, &j);
5	printf("%d %d", 10);
6	printf("%d %d", 1 % 2, 'I');
7	Int A,B;
8	Float rate = 3.7:

Fig 5. Identifying the errors



Test Your Understanding

Select valid function calls

void print_area(); ✗

sum= get_x(30) + get_y(40); ✓

print_volume(a, 10, c*2); ✓

send_signal(int x, int z); ✗

add(add(5,3), add(a,8)+10); ✓

printf("%d", add(34,98)); ✓

add(get_x(30), get_x(30)); ✓

div= int divide(34, 98); ✗

Fig 6. Snapshot of the Activity on Test your understanding

B. Assessment

The assessment in this course includes term-works, quizzes, debugging, code optimization and code review tests. This assessment is inherited from our previous work as in [2]. The term work evaluates students coding and problem solving skills. In the term work students are given on spot problem statement based on specific concept, where the student writes the test plan, codes for the given problem statement and executes the same [2].

A total of 100 marks are split in to In Semester Assessment (ISA) and End Semester Assessment (ESA) with 80 and 20 marks respectively. The term works including other tests and quizzes are assigned 50 and 30 marks respectively which are part of ISA. Here the rubrics pattern and explanation of rubrics for ESA 2015-16 are shown in Fig 1.

IV. RESULTS AND DISCUSSIONS

The performance of students in Hands-on approach (current year 2015-16) is compared with the traditional teaching approach (previous year 2014-15) as shown in Fig 7 and Fig 8.

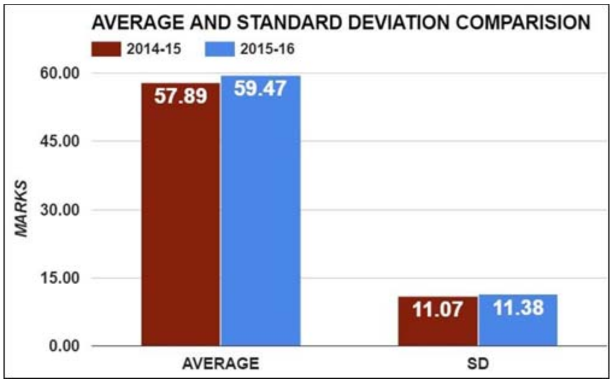


Fig 7. Average and standard deviation comparisons

There is improvement in average score for the current year as shown in the Fig 7. But SD (standard deviation) got slightly increased. When the results are compared grade wise, improvement in higher grades S, A are observed. But the number of failures increased in the current year as shown in Fig 8.

All these observations indicate that there is overall improvement in the performance of students in integrated laboratory class. Although there is no huge difference in performance of students, it can be improved drastically if written assessments are conducted. i.e. since in our case study of integrating lab in class, assessment is more focused on testing the problem solving and practical(coding) skills of students. But there is no written assessments conducted that tests the conceptual understanding of students. Due to this loophole, students didn't bother to read textbooks and rather relayed on class-room, hand-on, demonstration and programming materials given to them. The other assessment conducted is quiz, but it cannot test comprehending skills of students. Thus adding written assessment to our approach can hugely improve the performance of students.

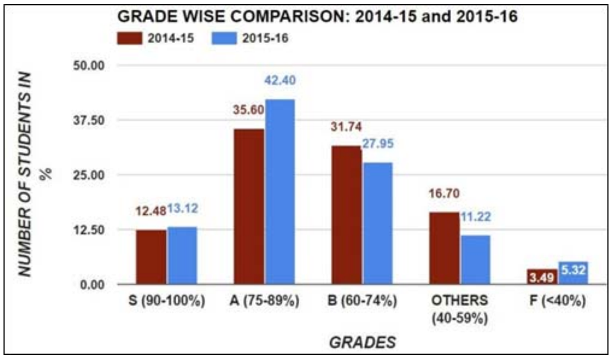


Fig 8. Grade wise comparison: 2014 and 2015

A. Students Feedback

The Table 1 gives the students feedback in which 117 students participated. The hand-on programming requires the use of technology. In this case study, LCD projectors and PPT (power point presentations used) are used. Although many students are happy with use PPT, few student's still feel blackboard usage helps them learn better. It is also observed that on-spot coding of programs in hands-on session improved students learning and their logical thinking (evident from performance shown in Table 1). Hands-on sessions increased the interaction between teachers and students within the classroom. It also kept students engaged and created interest about the course among the students. All these observations are evident from the student's feedback questions 2A, 2B, 2C and 2D given in Table 1.

TABLE I. STUDENTS FEEDBACK RATINGS

	Feedback in %				
	I strongly agree	**I agree**	**neutral**	**I disagree**	**I strongly disagree**
1A. Powerpoint presentations(ppt) used during teaching helped in better learning	33.3	38.5	20.5	4.3	3.4
1B. Programs given with solutions as homework helped in better learning of the concepts.	36.8	39.3	13.7	8.5	1.7
2A. Hands-on helped you develop	37.6	41	15.4	2.6	3.4

interest about the course.					
2B. Hands-on helped you in better learning.	40.2	41	10.3	4.3	4.3
2C. Hands-on increased interaction with teacher.	29.9	44.4	14.5	7.7	3.4
2D. Hands-on kept you engaged in laboratory.	35.9	43.6	11.1	6.8	2.6
2E. It was not easy for you to switch from hands-on mode to listening mode when next topic is taught.	22.2	32.5	32.5	10.3	2.6
2F. Hands-on activities were easy and you could complete them much earlier than your friends.	24.8	34.2	31.6	8.5	0.9
2G. Three hours session is long and was difficult to learn many topics in one session.	29.1	32.5	24.8	6.8	6.8
2H. Enough time was spent in laboratory to make you understand a concept.	23.9	44.4	18.8	8.5	4.3

The feedback question 2E, 2F, 2G and 2H gives the extent of challenges associated with hands-on programming teaching. During the hands-on sessions, a small lecture is followed by coding. Then after coding, again a small lecture is given on next topic. But students are still engaged and struck in the coding and they don't quickly switch to listening mode for lecture given on next topic. Also, many of the students could complete hands-on tasks quickly while others are still struck in coding. During this, teachers are also engaged in helping the students who are struck. Then the rest of students become passive and feels bored in the class.

The hands-on sessions are usually of longer durations. In this case study, each topic taught in the first half of the semesters consumed more duration. Eventually extra classes are taken at the end where lot of syllabus is covered in single session. But this proved to be burden on students since they didn't get sufficient time to react and reflect on the topics being taught.

Considering the teacher's experience, students' performance and feedback, following benefits and challenges are observed through this case study.

Benefits of Hands-on programming teaching:

i. Hands-on improved students learning and logical thinking required for programming.

ii. Hands-on increased the interactions between teachers and students. i.e students queries are addressed in classroom and teachers could get immediate feedback about the learning that happened in a session.

iii. The initial learning pace of students may be slow, but it gradually increases with number of classes.

Since this is first attempt of integrating laboratory class and hands-on programming, many challenges are observed which are as follows.

Challenges of Hands-on programming teaching:

i. Hands-on consumes more time than traditional class in teaching a topic and each session will be lengthier.

ii. It is difficult for students to switch between hands-on and lecture mode.

iii. Students will become passive when they can complete hands-on task earlier than peers. It leads to boredom.

iv. Hands-on programming hugely depends on technology and resources like power, computer devices, network, projectors, audio equipment's and enough number of instructors.

v. Maintaining continuity and revising can be difficult if there is a gap of 2-3 days between each session.

Following are the few comments given by students

"More than enough efforts were made by the course teacher, which has brought it's fruits. Though being a complete fresher to the course, every concept thought was easily understandable. Enough time was given to every program. One of the best taught course in our semester. Thank you sir."

"I hardly concentrated 1 hour out of 3 hours in theory classes , it is hectic to concentrate for 3hours, even if we forcibly concentrate we can't get the concept."

"Good teaching but more detailed should be taught overall good support from all teachers"

"It was very effective and useful... Course instructor has been the best guide and helped in learning more about the course as well as the outside questions n knowledge was given..."

V. Conclusions

The introductory programming course is conducted by integrating laboratory and class for the first time at our university. The planning and design of the course is based on the literature survey, experience of teachers and similar course integration done for the higher semester. At the end, most of expected results are obtained but there are surprises and discovery of challenges associated with hands-on programming.

The end results showed improvement in students' performance which developed problem solving skills and logical thinking in students. One important observation is that, this approach increased coding skills of students. i.e students are able to write syntax error free programs. But the performance in problem solving skills is satisfactory which can be improved by including assessments on conceptual understanding. i.e a written test is missing and quiz is taken instead of it. But in next venture quiz will be replaced by written test.

This approach improved laboratory skills and number of students scoring higher grades have been increased. i.e integration of class and lab helped to students to connect well between conceptual understanding and laboratory skills.

Students also expressed that course is well conducted and helped them learn better. Hand-on experience motivated them and developed interest about the course.

In this case study, the number of credits for course was reduced to 3 with Lecture-Tutorial-Practical: 0-0-3. While in the previous year, the same course was of 7 credits with separate class and laboratory. However the syllabus was not compromised. Also hands-on sessions consumed more time than traditional teaching. Due to these reasons extra classes are conducted to cover-up the syllabus. This raises the questions like

i. How to plan hands-on sessions?
ii. How much syllabus and which part of syllabus to be covered with hands-on sessions?
iii. Whether go ahead with full hands-on teaching or partial hands-on teaching for programming courses?

There are also many challenges of hands-on programming faced in this case study which are already discussed in Results section. These challenges throw up new research questions and further work can be carried out to give solutions.

Finally hands-on programming may increase the performance of students but it definitely ensures that learning happens in the class room. Once course designers and teachers get experienced and develop thorough understanding of hands-on programming science, they can plan to upgrade the course with studio based learning.

REFERENCES

[1] Handur, Vidya S., Jayalakshmi G. Naragund, and Priyadarshini D. Kalwad. "SUCC Tool Kit: An Activity in Data Structure Laboratory." In *Proceedings of the International Conference on Transformations in Engineering Education*, pp. 395-404. Springer India, 2015.

[2] Handur, Vidya S., Priyadarshini D. Kalwad, Nagaratna Yaligar, Vishwanath G. Garagad, and Manjula K. Pawar. "An activity based learning: C programming." In *MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on*, pp. 310-314. IEEE, 2015.

[3] Benander, Alan, Barbara Benander, and Janche Sang. "Factors related to the difficulty of learning to program in Java—an empirical study of non-novice programmers." *Information and Software Technology* 46, no. 2 (2004): 99-107.

[4] Milne, Iain, and Glenn Rowe. "Difficulties in learning and teaching programming—views of students and tutors." *Education and Information technologies* 7, no. 1 (2002): 55-66.

[5] Barak, Miri, Judson Harward, George Kocur, and Steven Lerman. "Transforming an introductory programming course: From lectures to active learning via wireless laptops." *Journal of Science Education and Technology*16, no. 4 (2007): 325-336.

[6] Bruner, Jerome S. *Acts of meaning*. Vol. 3. Harvard University Press, 1990.

[7] Von Glasersfeld, Ernst. *Radical Constructivism: A Way of Knowing and Learning. Studies in Mathematics Education Series: 6*. Falmer Press, Taylor & Francis Inc., 1900 Frost Road, Suite 101, Bristol, PA 19007., 1995.

[8] Byrnes, James P. *Cognitive development and learning in instructional contexts*. Allyn & Bacon, 2001.

[9] Haury, David L., and Peter Rillero. "Perspectives of Hands-On Science Teaching." (1994).

[10] Lumpe, Andrew T., and J. Steven Oliver. "Dimensions of hands-on science."*The American Biology Teacher* (1991): 345-348.

[11] Gucwa, Mr Kevin James, and Harry H. Cheng. "RoboSim for Integrated Computing and STEM Education." *age* 24 (2014): 1.

[12] Donivan, Marilee. "A Dynamic Duo Takes on Science." *Science and Children*31, no. 2 (1993): 29-32.

[13] Tobin, Kenneth. "Research on science laboratory activities: In pursuit of better questions and answers to improve learning." *School Science and Mathematics* 90, no. 5 (1990): 403-418.

[14] Penick, John E., and Robert E. Yager. "Learning from excellence: Some elementary exemplars." *Journal of Elementary Science Education* 5, no. 1 (1993): 1-9.

[15] Shymansky, James A., and John E. Penick. "Teacher behavior does make a difference in hands-on science classrooms." *School Science and Mathematics* 81, no. 5 (1981): 412-422.

[16] Brownlie, Faye, and Judith King. *Learning in safe schools*. Pembroke Publishers Limited, 2011.

[17] Bergin, David A. "Influences on classroom interest." *Educational psychologist* 34, no. 2 (1999): 87-98.

[18] Krapp, Andreas. "Basic needs and the development of interest and intrinsic motivational orientations." *Learning and Instruction* 15, no. 5 (2005): 381-395.

[19] U. Schiefele, "Interest, learning, and motivation," Educational psychologist,vol. 26, no. 3-4, pp. 299–323, 1991

[20] Hidi, Suzanne, and K. Ann Renninger. "The four-phase model of interest development." *Educational psychologist* 41, no. 2 (2006): 111-127.

[21] Krapp, U. Schiefele, and I. Schreyer, "meta-analysis of the relationship between interest and academic achievement," Journal f ”u r Developmental Psychology and p ¨a pedagogical psychology, vol. 10, no. 2, pp. 120–148, 1993

[22] Renner, John W., Michael R. Abraham, and Howard H. Birnie. "Secondary school students' beliefs about the physics laboratory." *Science Education*69, no. 5 (1985): 649-663.

[23] Hebbar, Santhosh, Praveenraj Pattar, and Varaprasad Golla. "A Mobile ZigBee Module in a Traffic Control System." *IEEE Potentials* 35, no. 1 (2016): 19-23.

[24] Holstermann, Nina, Dietmar Grube, and Susanne Bögeholz. "Hands-on activities and their influence on students' interest." *Research in Science Education* 40, no. 5 (2010): 743-757.

[25] Hegarty-Hazel, Elizabeth. *Student laboratory and the science curriculum*. Routledge, 1990.

[26] Canfield, Stephen, Sheikh Ghafoor, and Mohamed Abdelrahman. "Enhancing the programming experience for first-year engineering students through hands-on integrated computer experiences." *Journal of STEM Education: Innovations and Research* 13, no. 4 (2012): 43.