

Activity Based Programming Learning

Dr. Amit Lathigara¹, Dr. Paresh Tanna², Dr. Nirav Bhatt³

¹School of Engineering, RK University, Gujarat, India

²School of Engineering, RK University, Gujarat, India

³School of Engineering, RK University, Gujarat, India

¹amit.lathigara@rku.ac.in

²paresh.tanna@rku.ac.in

³nirav.bhatt@rku.ac.in

Abstract: Computer programming is the essential skill in today's era and students are learning computer programming from their school education and also for students who have been enrolled in engineering program branches like Computer Engineering and Information Technology, computer programming is the foundation for them. Students following traditional paper-pen based approach for the examinations are having remembering skills and generally these skills become the base for them to clear any examinations. In this study, a series of unique activities are performed in regular sessions and learning outcomes are measured and compared with traditional sessions. Computer programming has been taught by various instructors since many years and as an outcome unique instructional pedagogy evolved from time to time with the help of regular feedback from all the stockholders. Computer programming is interesting and easy to understand if sessions are executed in a unique way through active learning skills. Active learning is also playing a crucial role in this, and hands on sessions helped a lot to achieve course outcomes. Here, to analyse the effect of activity-based programming learning approach for the specified subject, the result of B.Tech. (CE and IT branches) second semester students (A.Y. 2018-19 and 2019-20) are considered with different familiar parameters. Students' feedback is taken at the end of course study. Find the word, crossword, code magnet, half cooking code, long exercise, pool puzzle, troubleshooting, predict the output, match two sides, program analysis are a set of activities tried throughout the course during every session. Assessment pattern is also changed effectively and instead of a traditional style pen-paper based sheet is converted into an activity sheet. Continuous assessment is practiced covering all components of the exam including end semester examination. Results found with traditional method-based examination are compared with results found with activity-based programming learning methodology and found that a greater number of students had secured more than over average score with the proposed method. This comparison shows that students are more distinguishable and curious about solving activity-based problems instead of robbing up theory concepts and writing the answers with traditional pen paper strategy. Also, student feedback is found quite satisfactory. It has been observed that through activity-based learning, student retention level is quite high and their learning skills and understanding level improved significantly. Active learning techniques perform an

essential role for accomplishing the student's participation and contribution towards learning. Logical and technical skills for engineering students can be developed easily with active participation i.e. through active learning.

Keywords: Activity based learning, Continuous assessment, Programming tasks, Find the word, Crossword, Code magnet, Half cooking code, Long exercise, Pool puzzle, Troubleshooting, Predict the output, Match two sides, Program analysis

1. Introduction

Generally, engineering students belong to various regions and having different background and many of them are having communication skills issues specially in first year of engineering. Also, they may have followed traditional paper-pen based approach for their past examinations in schooling education. So, naturally remembering skills become the base for them to clear any examinations. To cope up with this situation, student's involvement during the sessions, plays a vital role rather than one-way communication by the instructors.

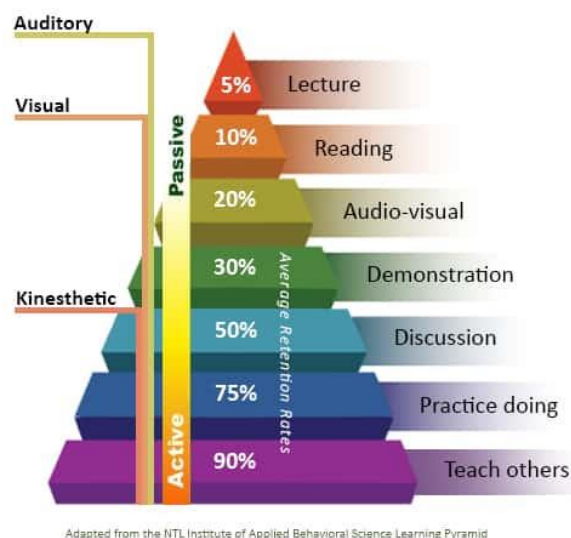


Fig 1. Learning Pyramid

Active learning techniques perform an essential role for accomplishing the student's participation and contribution

towards learning. The active learning techniques are students driven and the instructor becomes a facilitator to assist students with picking up information and abilities for initiating, applying, and rediscovering the information. From Fig. 1 of learning pyramid, it is noticeable that students can hold attention about 75% when they utilize the learning or practice it by doing given task at their level.

In fact, logical and technical skills matter a lot for engineering students specially computer engineering and information technology students. These skills can be developed easily with the programming skills-oriented subjects (Sharma, A., 2019).

Active learning methodologies are among the inspiring vehicles to involve and engage engineering students in project-based learning (R. Senthil, 2020). The proposed approach with active learning methodologies will help students to implement future tasks and provide adequate solutions to the problems based on active participation found in the real world.

2. Literature Review

Academicians and IT specialists focus more significantly on fundamentals of computer programming. Learning a structure of actual language statement will not contribute much to the overall development of skills related to critical thinking. Various existing approaches are reviewed to have more insights and matching with proposed approach.

In 2009, García-Mateos and Fernández-Alemán and Montoya-Dato et al. proposed Mooshak i.e. a web-based automatic judging system to receive and evaluate programs in Ada.

Limitation and Justification: This approach can be embraced to check with Ada programs only, and it is focusing on the entire program only whereas the proposed approach is manageable with several technical skills oriented parameters that can help to improve programming skills effectively.

In 2010, Zhang and Ke also proposed a plan for SQL language i.e. Paperless Examination System that could deal with both kinds of questions at practical level i.e. dealing with simple multiple-choice questions and processing subjective exam questions which involve judging programming language.

Limitation and Justification: It is limited to SQL language only i.e. with essential predefined queries and methods while proposed approach which is important with kind of technical skills tests to build up base for programming languages.

A web assessment tool known as EduPCR, is introduced to check the learning with a peer audit of programs written by distinctive students, share ideas and make proper suggestions (Wang, Y., 2012).

Limitation and Justification: This methodology utilizes the peer evaluation design; this may result in a one-sided score. While the proposed approach is completely instructor driven.

Here, authors worked for C++ programming language with more focus on library illustrations and semantic usage of language. They have specially focused on very specific concepts only with predefined programming language techniques (Ranade, A., 2016).

Limitation and Justification: This approach only focuses on basic concepts coverage of C++ language which is far away compared to the proposed approach which comprises of many activity-based learning techniques.

Restrepo-Calle, F., 2019 proposed a continuous assessment methodology for a computer programming course supported by an automatic assessment tool that applied to the practical programming exercises performed by the students.

Limitation and Justification: This approach suggests the corrections for the wrong programs. This moreover requests the entire program to be written for evaluation. Whereas proposed approach addresses on several parameters find the word, crossword, code magnet, half cooking code, long exercise, pool puzzle, troubleshooting, predict the output, match two sides, program analysis, etc.

The tutorials and examination based on troubleshooting & debugging may help students to learn programming language in an interactive way (Tanna, P., 2020).

Activity based methodology can be upgraded than traditional methodology for student's active contribution in learning and subsequently thus improving students learning (Patil, A., & Chavan, P, 2020)

Limitation and Justification: This approach as it were comes with limited types of questions where the proposed approach highlights many types of brainstorming questions by providing activity-based learning.

For the methodology utilized in all the different reviewed concerns, few of them are limited to whole programs or few are constrained to the specific programming language only. This suggests that a progressively feasible approach is required for developing and evaluating the student skills. Apart from recognized limitations/findings, the program quality and semantic comparability of a program with various parameters should be built up for programming skills improvement. Following this, a proposed approach is

planned for all sorts of programming languages with unique questions format.

3. Methodology

With the traditional pen paper approach, students are not assessed according to their abilities, however they are assessed based on their remembering skills like writing answers of recalling/understanding based questions in the given answer sheet. To defeat the issues found with traditional approach and to improve the students programming skills, one of the significant strategies is learning programming based on activities. Set of different type of activities are tried throughout the course like find the word, crossword, code magnet, half cooking code, long exercise, pool puzzle, troubleshooting, predict the output, match two sides and program analysis. Also, assessment is done based on the responses through different activity sheets and it has been completely different from traditional approach which helped a lot for supporting continuous assessment practice that covers all components of the exam including semester end examination. Relevant methods are elaborated to get the clear understanding for the proposed approach.

Find the word/content: Keyword exercise

In most of cases, facilitators are providing set of keywords in the beginning sessions of teaching programming languages. Instead of involving students to remember such set of words, find the word exercise keeps students engaged in listing words from given jumbled characters. Fig. 2 illustrates solved activity through all highlighted keywords of C programming language.

S	t	s	Y	e	p	t	e	r	d	d	c	e	f	d	H	t	r	u	r
i	u	n	S	i	a	n	e	d	e	u	e	o	e	c	I	n	t	n	o
g	a	t	S	e	s	c	a	f	e	u	n	t	e	C	s	o	n	o	g
n	t	c	L	x	a	w	a	r	e	s	n	i	p	f	U	t	o	f	i
e	u	o	E	S	i	u	r	e	d	i	w	t	y	g	N	o	t	j	s
b	e	a	A	t	i	g	t	t	e	s	i	j	s	v	R	r	e	f	t
d	e	a	A	t	k	e	o	d	g	t	n	t	o	S	T	u	l	i	s
s	t	a	t	e	c	r	t	u	p	n	e	k	r	o	S	s	i	n	r
c	n	t	Q	e	g	o	t	o	y	l	u	a	u	l	C	h	a	r	i
a	f	i	L	f	o	r	s	r	j	e	n	e	c	a	O	t	h	e	h
s	l	n	A	j	o	t	i	t	t	d	i	r	t	t	N	u	w	d	r
f	o	u	B	r	g	a	a	e	u	n	t	b	t	t	S	w	d	r	w
o	a	e	U	v	n	r	l	u	n	n	t	r	u	e	i	T	r	o	h
e	t	d	U	n	o	n	f	r	i	y	n	e	u	i	R	c	f	u	e
z	o	i	D	v	i	i	e	o	o	p	o	e	t	e	D	f	c	n	r
l	d	e	O	t	t	o	t	g	t	c	c	n	x	F	O	t	s	r	
s	c	f	N	f	f	n	e	y	n	d	o	u	b	i	E	n	u	m	u
e	r	c	D	i	t	o	r	u	r	e	f	o	i	w	L	d	r	i	t
c	c	a	T	o	q	f	t	y	p	e	d	e	f	h	S	e	t	g	e
h	a	u	L	a	w	h	i	j	e	d	e	n	t	i	E	l	s	n	

Fig. 2. Activity related to Find the Word

Half Cooking Code

Writing computer program is an art and it's all about logical skills. This activity is focusing more on development of logical skills. Instead of asking students to write a full computer program, first basic code will be provided, that

kick starts their thinking and ultimately core logical part will be assigned to them that they can develop/write. This way student can enjoy the overall development of full code through this activity i.e. known as Half Cooking Code. Fig. 3 shows the solved activity related to half cooking code technique.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int no;
    clrscr();
    printf("Enter any number for odd/event test");
    scanf("%d",&no);

    .....

    if(no%2==0)

    .....

        printf("\n %d is Even",no);

    .....

    else

    .....

        printf("\n %d is Odd",no);

    .....

    getch();
}
```

Fig. 3. Solved Half Cooking Code

Pool Puzzle

This activity allows student to think critically about main logic of source code. It is a fun activity where important part of code is hidden and set of related missing fraction of code is provided separately at end of code. Students need to find best suitable fraction of code from all provided set of options and needed to write at appropriate place in actual source code. Set of options as fraction of code is larger than actual needed code which makes student to think critically during placement of that fraction code in actual code. In this way, students understand the logic of code in well manner. Fig. 4 shows the unsolved activity related to the pool puzzle technique.

```
#include <stdio.h>
#include <conio.h>

void .....
{
    int ch,no1=15,no2=10;
    clrscr();
    printf("\n1-Addition");
    printf("\n2-Subtraction");
    printf("\nEnter your choice");
    scanf("%d",.....);
    .....(ch)
    {
        case ....:
            printf("\nAddition of .... and .... is ...., ...., ...., ....");
            .....;
        case ....:
            printf("\nSubtraction of %d from %d is %d", ...., ...., ....);
            .....;
        default:
            printf(".....");
    }
    getch();
}
```

&ch	1	case	2	break	continue
%f	no1+no2	break	no2	%d	switch
no1-no2				main()	no1
if	%d	no1	no2	%c	no1
%d			\nInvalid Choice		

Fig. 4. Unsolved activity related to Pool Puzzle

Troubleshooting

This activity involves student to debug the source code, resolve logical and syntax errors and make code functional as per requirements. As an exercise, source code with lots of logical and syntax errors are given to the students i.e. intentionally errors are kept in the main part of the source code to signify the importance of error solutions. Students will check the errors and solve the same to make source code further functional. Fig. 5 shows the unsolved activity related to troubleshooting technique.

```
//Program to print 1 to 10 on screen
#include (stdio.h)
#include (conio.h)

void Main()
{
    int no;
    clrscr();
    while(no>10)
    {
        printf("%D",No);
    }
    getch();
}
```

Fig. 5. Unsolved activity related to Troubleshooting.

Predict the output

This is a simple activity where functional source code is given to the students and they need to apply their logical skills to judge the correct output. In this way, they will strengthen their own logical skills. Fig. 6 shows the activity related to predict the output technique.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int x, y = 5, z = 5;
    x = y==z;
    printf("%d", x);
    getch();
}
```

Fig. 6. Activity related to Predict the Output

Match two sides

This is an activity useful to understand basic concepts of course. Two set of values are provided as per Fig. 7 and both are kept in two sides. Students need to match each value from left side to an appropriate value on the right side. It is a kind of brain mapping activity and useful to check knowledge level of basic concepts of course. Fig. 7 shows the solved activity related to match two sides technique.

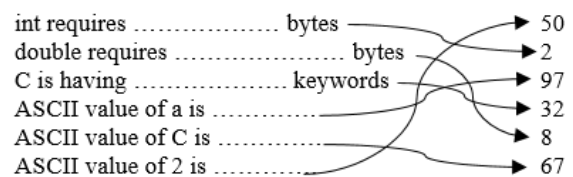


Fig. 7. Solved activity related to Match Two Sides

Program Analysis

This is an interesting activity, where student needs to justify the significance of every statement available in source code. It is serving a purpose of documentation also. Students are specifying the meaning and the need of each statement available in the source code and through this way, they can build their programming skills as well. Fig. 8 shows the solved activity related to program analysis technique.

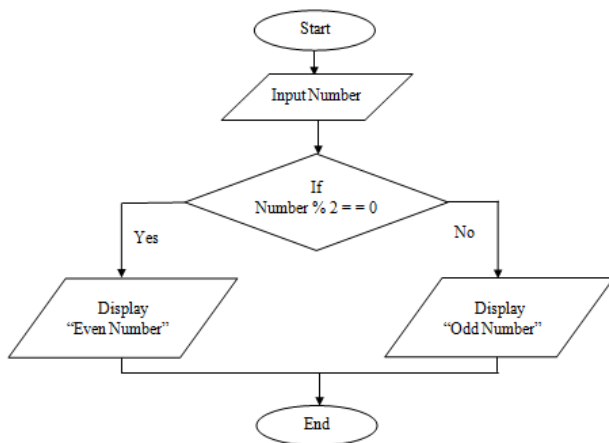
#include <stdio.h>	<u>Inclusion of header files to use its in-built functions in program</u>
#include <conio.h>	<u>Main method from which execution starts</u>
void main()	
{	
int no,fact=1,itr;	<u>Declaration of variables, 6 bytes of memory requirement</u>
clrscr();	<u>Clears output screen/erase previous output</u>
printf("Enter any integer no to find its factorial");	<u>Prints message on screen</u>
scanf("%d",&no);	<u>Prompt to the use to enter an integer number</u>
for(itr=no;itr>=1;itr--)	<u>Code block executes continuously starts from number till it reaches to 1 by decreasing current value by 1</u>
fact=fact*itr;	<u>Multiplies next number with previous result that ultimately holds result of factorial</u>
printf("\nFactorial of %d is %d",no,fact);	<u>Prints value of factorial on screen</u>
getch();	<u>Holds output temporary on screen, till user presses any key</u>
}	

Fig. 8. Solved activity related to Program Analysis

Long Exercise

This is an activity where students need to draw first a flow chart to solve the given problem. Flow chart is a graphical representation of actual source code. Once facilitator verifies the flow chart, then students need to write the algorithm based on flow chart i.e. a need for the source code development. Algorithm is the step wise descriptive instructions which are further needed to convert them into actual source code for the programming language. Again, facilitator verifies algorithm and based on algorithm, student needs to write actual source code. Fig. 9 shows all three components of long exercise technique.

Flowchart



Algorithm

Step 1: Start

Step 2: [Take Input] Read: N

Step 3: Check: If $N \% 2 == 0$ Then

Print : N is an Even Number.

Else

Print : N is an Odd Number.

Step 4: Exit

Program

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int no;
    clrscr();
    printf("Enter any integer no to find whether its odd or even");
    scanf("%d",&no);
    if(no%2==0)
        printf("\n%d is Even",no);
    else
        printf("\n%d is Odd",no);
    getch();
}
```

Fig. 9. Three Components of Long Exercise

Crossword

This activity is related to brain mapping task. Generally, students are filling crosswords in routine from their

childhood. In this technique, hints are given in across and down sections through which students need to find out the word needed to fill in crossword. Once they start filling crossword, they will get additional hints from filled portions continuously. Fig. 10 shows the unsolved crossword activity sheet.

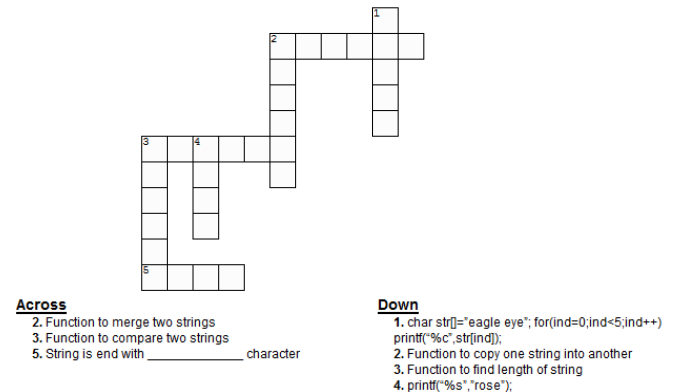


Fig. 10. Unsolved crossword activity

Feedback

For the proposed approach, the feedback was conducted from students' groups as shown in Table 4. This feedback indicates the students' understanding of the concepts and enthusiasm for the active learning approach towards FOCP programming subject.

Table: 4 Feedback for the proposed approach

Particular	Poor (%)	Fair (%)	Good (%)	Excellent (%)
Lectures are clear and organized with active learning.	2	5	42	51
The instructor respected students' perspectives to make it more attractive for technical aspects.	0	7	49	46
The instructor was timely in giving comments and reviewing students' work.	0	11	42	37
The instructor gave adequate extra help when needed.	0	2	43	55
The instructor encouraged class discussion.	0	4	45	51

5. Results and Discussion

Here, to analyse the effect of activity based programming learning approach for Fundamentals of Computer Programming (FOCP) subject, the result of B.Tech. (CE and IT branches) second semester students (A.Y. 2018-19 and 2019-20) are considered inside the Table 1 with different parameters (Ganesh, 2018). Result of the academic year (A.Y. 2018-2019) found like around 43% students who had secured more than over average score with traditional method-based examination whereas the result of the academic year (A.Y. 2019-2020) which was conducted with activity based programming learning methodology found like around 83% students had secured more than over average score i.e. showed up in Fig. 11 and 12. This comparison shows that students are more distinguishable and curious about solving activity-based problems instead of robbing up theory concepts and writing the answers with traditional pen paper strategy.

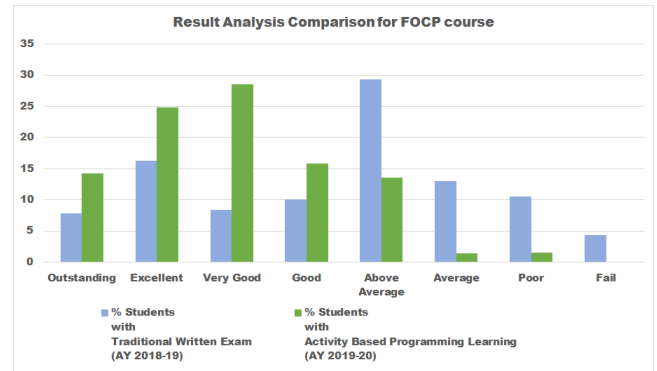


Fig. 11. Result Analysis Comparison (A)

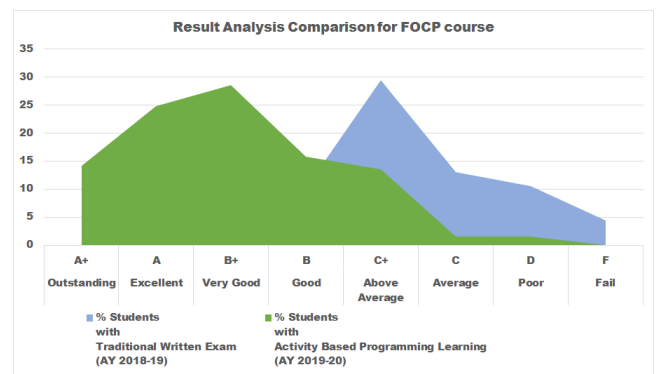


Fig. 12. Result Analysis Comparison (B)

Table: 1 Result Analysis Comparison for FOCP course

FOCP Result Analysis (For AY 2018-19 & 2019-20)			
Level	Grade	% Students with Traditional Written Exam (2018-19)	% Students with Activity Based Programming Learning (2019-20)
Outstanding	A+	7.82	14.21
Excellent	A	16.32	24.82
Very Good	B+	8.44	28.57
Good	B	10.13	15.83
Above Average	C+	29.38	13.57
Average	C	13.05	1.46
Poor	D	10.52	1.54
Fail	F	4.34	0.00

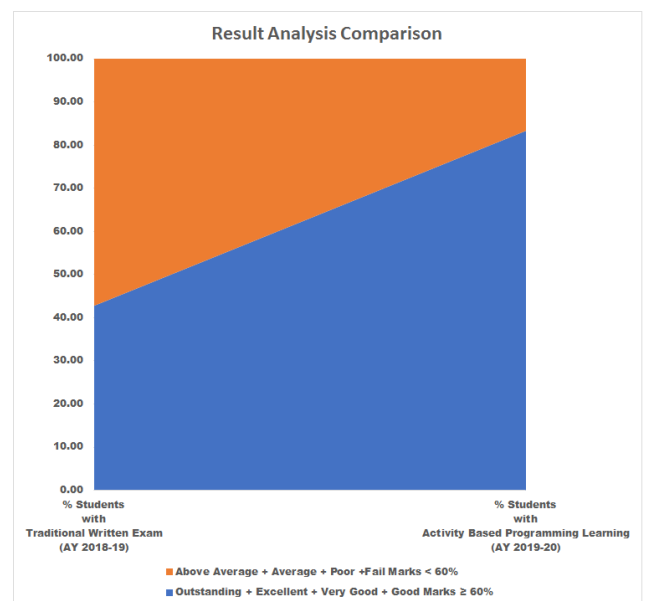


Fig. 13. Result Analysis Comparison between “ $\geq 60\%$ Students Group” & “ $< 60\%$ Students Group”

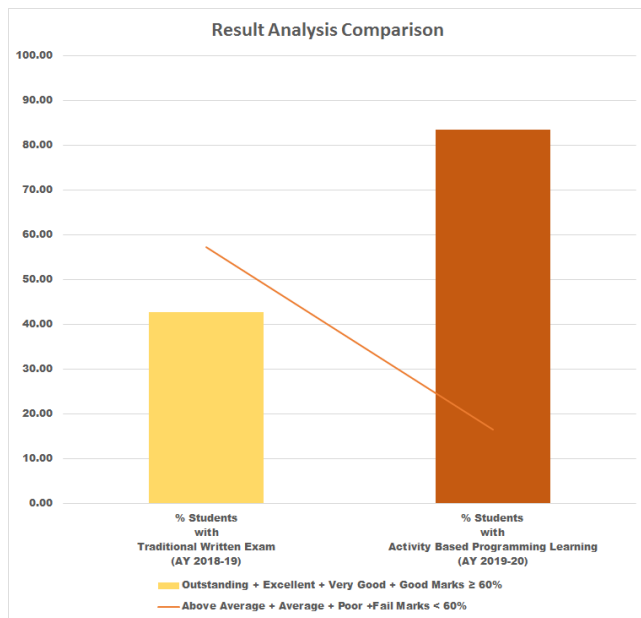


Fig. 14. Result Analysis Comparison between “ $\geq 60\%$ Students Group” & “ $< 60\%$ Students Group”

Both Fig. 13 and 14 highlight the increasing number of students with grades (A+, A, B+, B) compared to the traditional written approach.

The Control group and test group are set up to evaluate the effect of the proposed approach on student's knowledge. The control group i.e. traditional pen paper approach (A.Y. 2018-19 Exam Evaluation i.e. given in last second column of Table 1) and experimental group (AY 2019-20 Exam Evaluation i.e. given in last column) i.e. proposed approach are compared against each other in this experiment.

The descriptive analysis was utilized to illustrate the mean and the standard deviation of the score. Since the secured marks for traditional and proposed approach exams were collected for the same subject in two different years, the paired t-test was utilized to test the significant distinction for both exams' score. The paired t-test was conducted to test the hypothesis.

Table 2 shows the combined tests' measurements of traditional written approach and proposed approach. The smallest mean with value 10.68% is found in traditional written approach for level 1 (A+, A, B+, B) grades which suggests that students are exceptionally fragile to perform in traditional written exam for the programming course. On the other side, the proposed approach with the mean value of 20.86% shows that students had excellent knowledge for solving practical problems. However, the overall result shows that there is around 50% improvement with the proposed approach for level 1 (A+, A, B+, B) grades.

Table: 2 The Paired Samples Statistics of Traditional and Activity Based Programming Learning Approach

Level	Traditional Written Approach (%)		Activity Based Programming Learning Approach (%)	
	Mean	Standard Deviation	Mean	Standard Deviation
Outstanding + Excellent + Very Good + Good	10.68	3.89	20.86	6.94
Above Average + Average + Poor + Fail	14.32	10.68	4.14	6.33

In traditional written approach, the standard deviation is very scattered compared to the proposed approach for level - 1 grades. Whereas, on the other side both mean and standard deviations are higher for the traditional written approach for level - 2 (C+, C, D, F) grades which also shows the lower impact of traditional approach.

The hypothesis testing with paired sample t-test was utilized for this study. The results are anticipated to suggest higher mean value for activity-based programming learning approach than traditional written approach.

The null hypothesis H_0 : There's no distinction in mean for the proposed approach than the traditional written approach.

Elective hypothesis H_1 : There's a distinction in mean for the proposed approach than the traditional written approach.

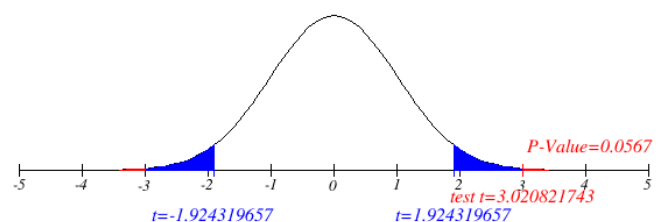


Fig. 15. Results of Paired Samples t-Test over Level – 1 Grades (Outstanding + Excellent + Very Good + Good)

Table: 3 Results of Paired Samples T-Test of Traditional and Activity Based Programming Learning Approach

Level	Paired Differences (%)		t-test result (2-tailed)
	Mean	Standard Deviation	
Outstanding + Excellent + Very Good + Good	8.66	3.68	0.056

Table 3 and Fig. 15 shows the findings of paired t-test for level-1 grades. At 10% significance level, all null hypotheses are rejected and t-test result i.e. P-value indicates that with proposed approach results has been improved undoubtedly compared to traditional written approach. In such a way, it can be concluded that there is a statistically remarkable contrast between mean score of the traditional written approach and proposed approach. Subsequently, it is noticeable that the proposed approach can upgrade the students' practical skills essentially.

6. Conclusions

The instructional exercises and examination based on activity-based programming learning approach may aid more to programming language learning in an intuitive way. Moreover, industry looks for students who are skilled with sufficient practical skills which may be the result of the proposed approach. Learning motivation from experience of practical work based on activity will strongly inspire the larger mass for engineering as a career choice.

References

- Montoya-Dato, F. J., Fernández-Alemán, J. L., & García-Mateos, G. (2009, June). An experience on Ada programming using on-line judging. In *International Conference on Reliable Software Technologies* (pp. 75-89). Springer, Berlin, Heidelberg.
- Zhang, G., & Ke, H. (2010, January). SQL paperless examination system design. In *2010 Second International Conference on Computer Modeling and Simulation* (Vol. 3, pp. 475-478). IEEE.
- Wang, Y., Li, H., Feng, Y., Jiang, Y., & Liu, Y. (2012). Assessment of programming language learning based on peer code review model: Implementation and experience report. *Computers & Education*, 59(2), 412-422.

Ranade, A. (2016). An Approach to Introductory Programming. *Journal of Engineering Education Transformations*, 29(3), 26-32. doi:10.16920/jeet/2016/v29i3/85205

Ganesh, K. E., & Pranisha, T. S. (2018). Enhancement of Learning Outcomes through Implementation of best practices in Teaching Learning Process: A case study. *Journal of Engineering Education Transformations*, 32(1), 12-14.

Sharma, A., & Arora, A. S. (2019). Active learning: An Instructional Technique for Improving Educational Practices. *Journal of Engineering Education Transformations*, 32(3), 36-39.

Restrepo-Calle, F., Ramírez Echeverry, J. J., & González, F. A. (2019). Continuous assessment in a computer programming course supported by a software tool. *Computer Applications in Engineering Education*, 27(1), 80-89.

Tanna, P., Bhatt, N., & Patel, S. (2020). An Innovative Approach for Learning and Evaluating Programming-Oriented Courses. *Journal of Engineering Education Transformations*, 33(3), 62-74.

Senthil, R. (2020). Enhancement of Engineering Education by Incorporating Active Learning Methodologies. *Journal of Engineering Education Transformations*, 34(1), 12-20. doi:10.16920/jeet/2020/v34i1/155008

Patil, A., & Chavan, P. (2020). ABDP: Activity Based Approach to Develop Students Conceptual Understanding of the Course Recycling and Regeneration Technology. *Journal of Engineering Education Transformations*, 33, 604-607. doi:10.16920/jeet/2020/v33i0/150129