Journal of Engineering Education Transformations,
Volume No 35, January 2022, Special issue, eISSN 2394-1707

Incorporating Visualization Tools and Active Learning Approach for Programming Courses

Prof. Ashwini Patil[1*], Prof. Dadaso Mane[2], Prof. Nitish Shinde[3]
[1,2,3] Dept. of Information Technology, Rajarambapu Institute of Technology, Rajaramnagar Sangli (MH), India

***Abstract*—** Programming languages are a vital domain for Computer and Information Technology engineering. The programming languages like C, C++, Java, Python, Mobile Application Development, and Web Technologies are important programming courses in CSE and IT curriculum. Understanding the programming courses includes a basic understanding of the syntax of programming language, logical & critical thinking to solve the problem, use of various editors to write and execute the program. Effective delivery of the programming course creates a great deal of influence on student’s interest in programming and software development. The students lose their interest in programming due to the inefficient delivery of these courses which leads to a gap in technical competency required by the IT sector.

The current Covid 19 pandemic exposed the limitation of traditional teaching methods for programming courses. Teaching such programming courses on online mode becomes a big challenge for faculty by considering the skillset required to inculcate amongst the students for learning programming courses. The primary objective of this paper is to overcome these challenges by the effective use of a pedagogical approach for teaching programming courses. The various active learning methods and tools are discussed in this paper that was used to teach the programming language. The result shows that by implementation of such methods & tools you can retain the interest of students in programming courses and thus increase their performance and improve the skillset required in IT industry.

***Keywords*—** Computer Programming, Activity-based learning, Visualization tools, logic building, Critical thinking, Continuous Assessment.

I. INTRODUCTION

Software developer, Application analyst, Software tester, System analyst, Web designer are various job profiles in the IT sector. Programming courses in CSE and IT curriculum plays important role in developing the technical competency required for such job profiles. From the learner’s point of view understanding programming language concepts and syntax is a challenging task. In physical mode, we have a direct connection to students, therefore we have a better understanding of the problems that students come across while learning programming. Teaching programming in online mode has become a challenging task for a teacher to ensure a joyful learning environment.

Programming language knowledge requires problem analysis, logical thinking, employing the use of algorithms & flowcharts to solve the problem, assembling the program using the proper syntax of programming language, program debugging skills to remove syntax & logical errors, and executing the program to get the desired output. Also, students should have program analysis skills to predict the output of the given program.

Only traditional teaching is not sufficient for programming courses to inculcate the above skills among the students.

While learning to program, students have several difficulties such as understanding how memory can be allocated to variables, how values of variables go on changing during execution, how to recognize and resolve the syntax & logical errors if any. To overcome these difficulties, it is essential to make a change in the teaching methodology and increase the use of many interactive tools for better delivery of programming courses.

While solving the problem using programming language, if students have the model on how programming problems can be solved then it will benefit them [1]. So it is important to incorporate the visual demonstration to the students about how the program works, how we can solve the problem using a programming language. The main goal of programming courses is to develop the ability of coding and debugging among students [9].

Concerning figure 1 of the learning pyramid/ cone of learning by the National Training Laboratory; it is clear that 75% of concepts get retained if students can practice it by doing and nearly 90% get retained if they teach others.

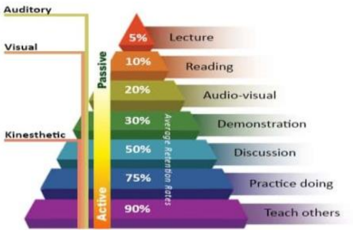


Fig. 1: Learning Pyramid

So doing the practice and hands-on practical session is very helpful for a better understanding of any course.

Active learning is an extremely important aspect of learning programming courses [2].

Active learning tools/techniques help to improve critical thinking and strong technical competency in their profession [6].

There are various active learning techniques used in Teaching and Learning the programming courses,

- ✓ Summary writing
- ✓ Collaborative Learning
- ✓ Reciprocal Teaching
- ✓ Problem based Learning
- ✓ Stump Your partner
- ✓ Use of visualization tools
- ✓ Pair Programming
- ✓ Short Quizzes
- ✓ Project-Based Learning
- ✓ Filliped classroom
- ✓ Group Discussion and Question Answer
- ✓ Brainstorming
- ✓ Prediction of the output of the given program
- ✓ Use of MOOCs Courses

We can make use of such active learning techniques for better delivery of programming courses. These techniques inculcate the computational thinking process among the students [7]. The active participation of students in the learning process is very important to improve the logical thinking and problem-solving ability of students, thus they can do programming using any programming language [3]. Using the pair programming technique, students can overcome the fear of programming by practicing programs with their friends, which makes learning more productive [8].

Project-Based Learning has proved to get better learner preservation, amplify the long-term interest, and increase presentation in design programming courses. Usually, while schooling a programming language, students have to be specified comprehensive instructions to arrive with improved understanding and problem-solving capability. The noteworthy purpose of PBL activity is to extend competent technological skills for students. The principle of such projects mainly lies in increasing student's perception in the actions of technical exploration and consciousness of the character and practices of technical investigation within their particular discipline. The programming concepts delivered are proposed to take the form of a mini-project. This procedure allows the students to work and feel like scientists. In practice, within the timescale of the B. Tech. program, it is a difficult job since a lot of students find this move from the usual process of learning to open-ended project-based learning hard. The students need grounding and holding up if they are to build a large amount of knowledge. The academic information gained by the students is inadequate when implementing a project [17][18].

In this paper, we have discussed active teaching-learning techniques and assessment methods used for Programming languages.

II. Teaching Methodology for Programming Courses

Understanding of any programming language includes various stages like an introduction to the concept, the syntax of programming language, understanding of problem statement, design the solution, implementation of the program, testing the program for syntax error, executing the program to get desired output with suitable test cases as shown in fig. 2.

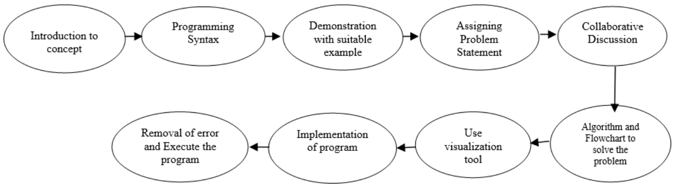
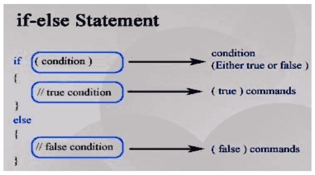

Introduction to concept
Programming Syntax
Demonstration with suitable example
Assigning Problem Statement
Collaborative Discussion
Algorithm and Flowchart to solve the problem
Use visualization tool
Implementation of program
Removal of error and Execute the program

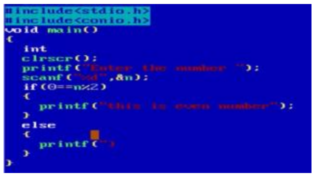
Fig. 2: Stages of Learning Programming Course

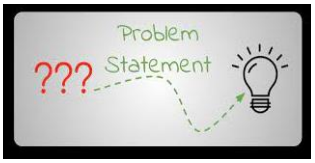
The stages to understand the programming language concepts are given in following table I,

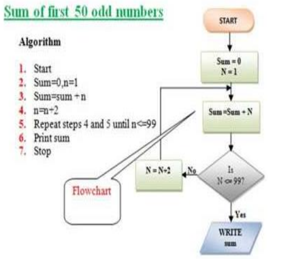
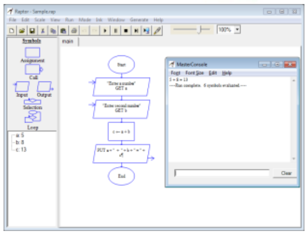
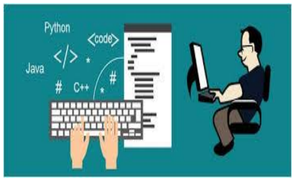
TABLE I
Stages of Learning Programming Course with an explanation.

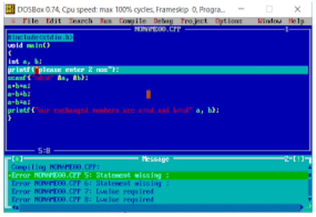
Introduction of concept		Explain the basic programming concept, and its need and application

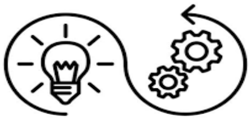
Programming syntax	 if-else Statement
	if
	(condition)
	condition (Either true or false)
	// true condition
	(true) commands
	else
	// false condition
	(false) commands
		Explain the programming syntax to implement that concept

Demonstration with suitable example	`#include<stdio.h>`
	`#include<conio.h>`
	`void main()`
	`{`
	`  int`
	`  clrscr();`
	`  printf("Enter the number ");`
	`  scanf("%d",&n);`
	`  if(0==n%2)`
	`  {`
	`    printf("this is even number");`
	`  }`
	`  else`
	`  {`
	`    printf("")`
	`  }`
	`}`
		Need to demonstrate how to implement that concept with suitable example

Assignment of a program statement		Once the theoretical concept, the syntax is clear to the students, we can assign the program statement to students
Collaborative discussion		The collaborative discussion was carried out among the students regarding how to solve the program

Design and devolvement for algorithm and flowchart to solve given program	 Sum of first 50 odd numbers Algorithm 1. Start 2. Sum=0,n=1 3. Sum=sum+n 4. n=n+2 5. Repeat steps 4 and 5 until n<=99 6. Print sum 7. Stop START Sum = 0 N = 1 Sum =Sum + N Is N <= 99? No N = N+2 Yes WRITE sum Flowchart	Then students need to write the algorithm (Solution Steps) and its visual representation i.e flowchart before actual implementation of the problem
Use of visualization tool to test solution	 Raptor - Sample.rap File Edit Scale View Run Mode Ink Window Generate Help 100% Symbols Assignment Call Input Output Selection Loop main Start "Enter a number" GET a "Enter second number" GET b c ← a + b PUT a+" + "+b+" = "+c¶ End MasterConsole Font Font Size Edit Help Clear a: 5 b: 8 c: 13 5 + 8 = 13 ----Run complete. 6 symbols evaluated.----	We can also make use of some visualization tool like a raptor to check the correctness for the solution.
Implementation of program		Students are asked to use an editor to write and implement the program

The removal of syntax error (if any)	 DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
	File Edit Search Run Compile Debug Project Options Window Help
	NONAME00.CPP
	5:8
	Message
	#include<stdio.h>
	void main()
	{
	int a, b;
	printf("please enter 2 nos"):
	scanf("%d%d" &a, &b);
	a+b=a;
	a-b+b;
	a-b+a;
	printf("our exchanged numbers are a=%d and b=%d" a, b);
	}
	Compiling NONAME00.CPP:
	•Error NONAME00.CPP 5: Statement missing ;
	Error NONAME00.CPP 6: Statement missing ;
	Error NONAME00.CPP 7: Lvalue required
	Error NONAME00.CPP 8: Lvalue required
		Students need to check the syntax errors if any in the program and correct those errors

Execute the program for all possible test cases		Students need to execute the program for all possible test cases to get desired output.

The students can achieve the expected outcomes such as problem-solving, designing algorithms & flowcharts, implementation of programs [7].

III. Literature Review

Lathigara, A., Tanna et al.(2021), the programming is interesting and easy to understand if it has been taught in a unique way using active learning skills [10]. Authors have used different types of activities like crossword, code magnet, half cooking code, long exercise, pool puzzle, troubleshooting, predict the output, and match two sides and program analysis as active learning techniques in computer programming courses. The assessment was also done based on responses to various activities. They proved that students' retention level, learning skills get improved as the output of these activities.

Thorat, S. A., & Kshirsagar, D. P. (2021), suggested the different activities for various problems faced by First Year engineering students for a computer programming courses. They observe that students face many problems like anxiety about computer systems, first-time programming, casual approach towards syntax, logic development issue while implementing the C program. This problem is solved by different activities like assignment of small activities to handle the computer, topic-wise short code challenges, writing algorithm and drawing flowchart, and code debugging. They observed that the result and overall performance of students get improved due to such kind of active learning technique [11].

Reddy, P. C. (2015), analyzed that making use of software tools to see the intermediate values of a variable during program execution as well as incremental programming approach helps to improve the programming ability of students [12].

Doddamani, S. T. (2018), suggested the use of animation-based software to demonstrate the concept, referral programming, and project-based learning. Effective use of these activities helps students for a better understanding of programming concepts and thus improve their programming skills [13].

Liu, S et al. (2013), mentioned that the program demonstration and practical-based teaching produce a good impact on students' study [14].

Handur, V. S et al. (2015), used various active learning methodologies like the use of coding standards, test-driven programming, pair programming, code optimization, interpretation, and debugging. This journey of students through various activities helps to increase the learning ability, problem-solving ability and develops the programming skills of students [15].

Blended learning adopts the finest essentials of established instruction with those of distance instruction. Blended learning is a mixture of established schooling methods (face to face) and online teaching based on education platforms. Learning in the classroom benefits individual interaction while online learning allows the student to plan in time and space along with their planned course activities and the pace of learning. However, it is for the instructor to decide which instructive essentials should be discussed in the division or online. The beliefs of this education mode are moderately simple: to capitalize on the pros of face-to-face and distance instruction [16].

Project-Based Learning (PBL) classically desires critical thinking, investigative capability, group effort, and unusual forms of communication. To accommodate the technological standards and produce the finer effort, it is indispensable for students to do notably more than memorize the mutual information. Python programming typically being a programming language required an unconventional instruction style to communicate to the students the essential technical skills and problem-solving capability. Hence, a methodical approach encouraging the students to comprehend the syntax and apply the same to provide simple solutions to local problems was adopted [18].

IV. Challenges while Teaching Programming Courses

The students who are learning programming first time have a lot of fear in their minds and this will lose their confidence level while doing programs.

Teaching programming courses has a lot of challenges, especially for first-year C programming courses. The first engineering students come from different backgrounds, some of them are having a problem with communication skills, less confidence, unfamiliar with friends, having traditional learning experience using paper -pen, less exposure to computer skills, and having just remembering skills rather than applying skill[10][11].

At the first-year level, some students are having anxiety about Computer Systems [11]. They are facing a problem right from starting of computer, installing software on the computer, use of Input-output devices. This makes it a challenging task for a teacher to prepare the students to deal with such kinds of issues and then teach them programming concepts.

Following are the common challenges or issues that the students face while learning programming courses:

1) Difficulty in handling computer system
2) Recalling the syntax of programming language
3) Development of logic to solve the given problem by applying programming concepts
4) Analytical skills to detect the syntax and logical errors in the program.
5) Analytical skill to predict the output of a given program

The students find it hard to gather the essential data and are unconscious of what is to be achieved. Many students have inadequate consciousness of the connection between theory and data. They are unaware that results can be interpreted from theory and vice-versa. As a result, they come across hard to understand the outcome which they get. These consequences lead students to experience discouragement. [17][18]

V. Active learning techniques for programming courses

In this paper, we have discussed the active learning techniques for 2 programming courses i.e C programming (at First yr level), Python programming (at Second yr level) of the CSIT curriculum. These techniques are also applicable for other programming courses.

❖ ***<u>Active learning technique for C Programming Course:</u>***

As this course is at a first-year engineering level, we need to focus on appropriate active learning methods to tackle the challenges mentioned above.

Following are some active learning techniques that we have used in the C Programming course:

1) Summary Writing:

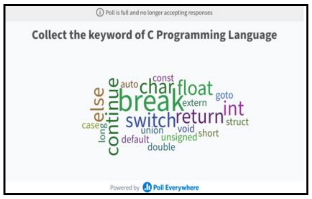
At the end of every lecture, we asked the students to write the summary of programming syntax taught during the lecture. This will improve their memorizing skills.

2) Demonstration of concepts:

In programming courses, teaching the theoretical concepts is not enough, we give more focus on the demonstration of concept on different editors like DevC++ or TurboC by demonstrating the actual implementation of the concepts.

3) Word cloud exercise:

The Word Cloud option from PoolEverywhere.com is used during the lecture. During this activity, the students are asked to post the new words that they learned during the lecture on the word cloud.



ⓘ Poll is full and no longer accepting responses

Collect the keyword of C Programming Language

Powered by Poll Everywhere

E.g Suppose the topic of the lecture is Keyword in C programming.

Then at end of the lecture, students simply post the different keywords that they learn from C Programming. This will increase their remembering skills.

4) Short Quizzes through Polls or Kahoot:

The student's attention span for the lecture is 10 to 15 min. So it is very essential to engage the student by doing some activity. So we have planned Short Quizzes on online interactive tools such as Polleverywhere and Khoot during the lecture to ensure the learning of concepts by students [19][20].

The MKCL clicker software is also used for the conduction of quizzes during classroom teaching. Using the MKCL clicker software it is possible to conduct the quiz in the classroom and collect the pool from the student. Students can use their mobiles to answer the questions.

Such kinds of quizzes and polls during lectures are used to get feedback about the understanding level of the student.

5) Stump your partner:

During the classroom teaching, we have used Stump your partner activity as shown in fig.3, where students are supposed to give one problem statement based on taught concepts, to the student sitting next to him/her, and so on. Then the students will be given some time to write a C program on the problem that can be given by his/her partner.

At the end of this activity, the teacher will ask any of the students to come in front and write the code on the board to explain it. This kind of activity helps to build critical thinking among students about the applicability of the concept to solve the problem.

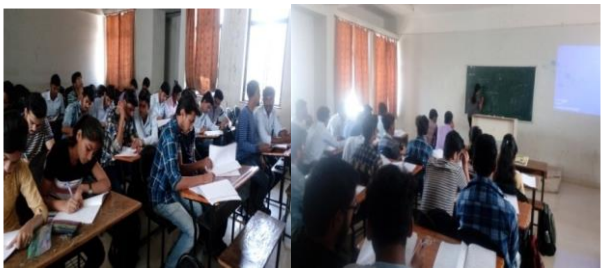
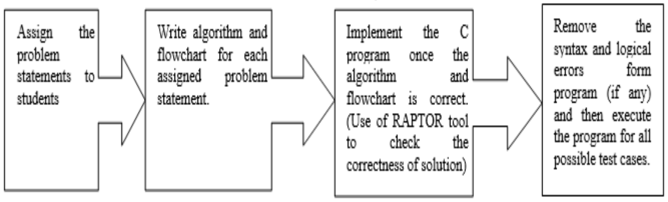


Fig. 3: Snap of stump your partner activity during class

6) *Problem-based learning by writing Algorithm and Flowchart:*

For the programming course, problem-based learning is very important. For the C Programming course, we have a total of 10 experiments for the lab sessions. During the lab sessions, we are given 14-15 problem statements to the students of different complexity levels for each experiment. Two lab sessions (i.e 4 hr) were given to students to implement this code for this given problem statement.

Steps in problem-based learning for the C Programming course at FY Level are shown in fig. 4:


Assign the problem statements to students

Write algorithm and flowchart for each assigned problem statement.

Implement the C program once the algorithm and flowchart is correct. (Use of RAPTOR tool to check the correctness of solution)

Remove the syntax and logical errors form program (if any) and then execute the program for all possible test cases.

Fig. 4: Steps of Problem based learning

7) *Use of visualization tools like Raptor, Python Tutor*

a) Raptor Tool:

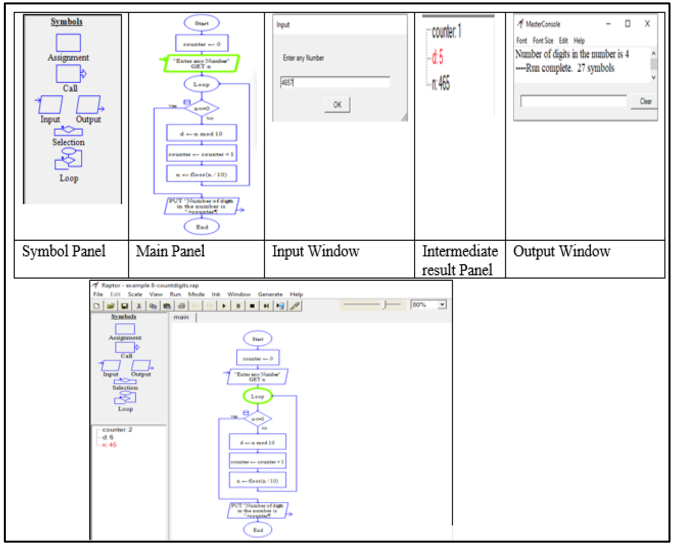
Rapid Algorithmic Prototyping Tool for Ordered Reasoning (RAPTOR) is a freely available tool designed by Martin C. Carlisle, Terry Wilson, Jeff Humphries, and Jason Moore. This tool is used for designing and executing the flowchart. Students can visualize the execution of the flowchart using the RAPTOR tool. It has a very simple GUI which has a drag and drop features [21].

The Raptor following windows:

1) Symbol panel: It consists of different symbols used to draw the flowchart.
2) Main panel: This is the main drawing section where the students can design the flowchart.
3) Intermediate result panel: It shows intermediate values of a variable while execution.
4) Input window: It is used to give input.
5) Output window: Used to display the output

Fig. 5 shows the use of the raptor tool:

Problem statement: counting the number of digits in the given number.



```
Number of digits in the number is 4
----Run complete.  27 symbols
```

Fig. 5: Use of Raptor Tool

Benefits of using raptor tool:

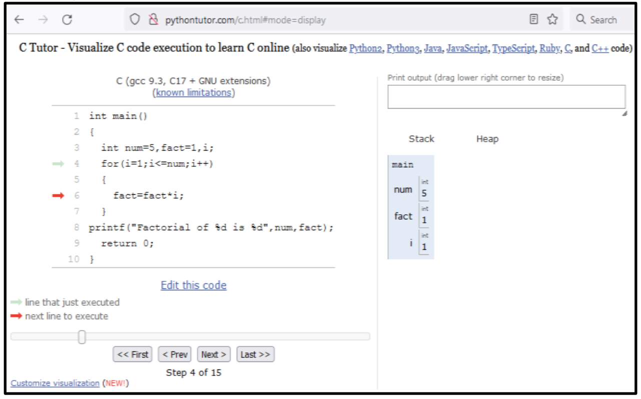
Students can

- Draw the flowchart for the given problem
- Execute the flowchart
- Visualize the execution of the flowchart, understand how the values of variable get changes and see the final output
- Verify the correctness of the solution by executing the designed flowchart.

b) Python Tutor:

Students can write/implement the C code, but sometimes they are unable to explain the working flow of the implemented program. They cannot able to explain how memory is allocated to variables, array; how values of variable get changed during the execution of the program; how function call get occurs; how the structure works; how pointer works.

The use of visualization tools helps the students to understand the concepts in a better way. Python Tutor is one of the freely available visualization tool through which students can understand what exactly happens when the code gets executed line by line. It is web-based tool that allows writing code in various languages like C, C++, Java, JavaScript, Python, Ruby, and visualize the execution of code line by line [22].


pythontutor.com/c.html#mode=display
Search
C Tutor - Visualize C code execution to learn C online (also visualize Python2, Python3, Java, JavaScript, TypeScript, Ruby, C, and C++ code)
C (gcc 9.3, C17 + GNU extensions)
(known limitations)
Print output (drag lower right corner to resize)
Stack
Heap
main
num
fact
i
int
Edit this code
line that just executed
next line to execute
<< First
< Prev
Next >
Last >>
Step 4 of 15
Customize visualization (NEW!)

```
int main()
{
  int num=5,fact=1,i;
  for(i=1;i<=num;i++)
  {
    fact=fact*i;
  }
printf("Factorial of %d is %d",num,fact);
  return 0;
}
```

num 5, fact 1, i 1

Fig. 6: Use of Python Tutor

Problem Statement: Find the factorial of the number.

Fig. 6 shows the screenshot of the python tutor visualization tool, which consist of the main program window (Where we can write our code), output window (where output gets displayed), memory window- heap, and stack (where it shows how memory is allocated to the variables used in the program) and arrow (Red arrow: shows statement going to execute and Green arrow: show statement get executed). There are a few more options shown at the bottom like First, Last, Prev, and Next through which we can control the flow of execution.

As you can see in the above code, the loop was used to find the factorial of a number, where the value of iteration variable ‘i’ goes on changing from 1 to number and you will get the final output displayed on the output window as shown in fig.7.

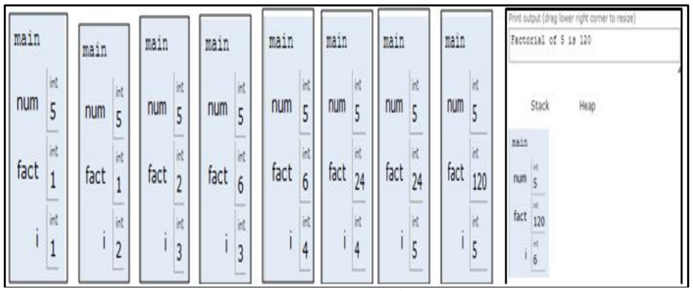


Fig. 7: Memory window snap of Python Tutor

Using the Python Tutor we can able to see the visualization of how the program gets executed. So using such a tool helps the students to understand the concepts easily and increase their interest in learning the programming languages.

8) Pair Programming for a lab session:

In Pair programming, two students work in pairs during the lab session to solve the given problem statements. This technique is very helpful to FY students because programming is new to them and they have a lot of fear in their minds while working during lab. Sometimes they are not able to recall the syntax.

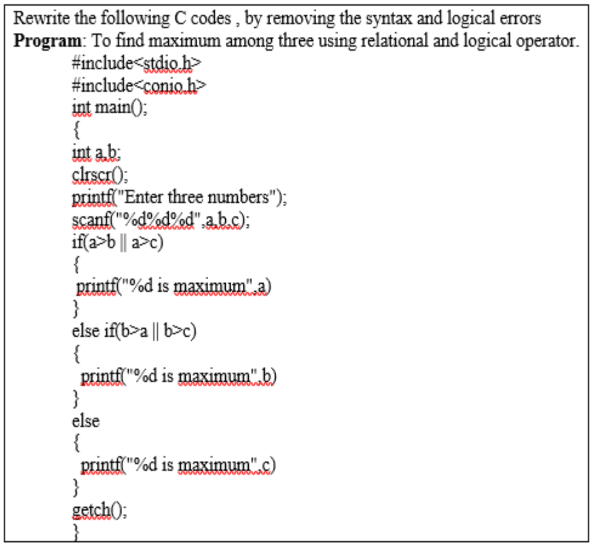
In pair programming, two students who perform the same experiment sit together on a single computer. Both of them write an algorithm and draw the flowchart with discussion. For implementing the code, one student can write the code on the computer (Programmer role) and other students help her to implement the code with syntax (Helper role). Discussion among these students helps to build the confidence level, remove programming fear from their mind and they can able to complete the given task.

There is role switching in pair programming i.e after the first program gets completed, the role of programmer and helper get changed. So both of them get an equal chance to write and execute the program on the computer.

9) Program analysis to detect the syntax and logical errors:

In this kind of activity, the readymade C program was given to students and asked them to detect the syntax and logical errors in the given program. This will improve the analytical and logical skill of the students.

Fig. 8 shows one example of this activity.



Rewrite the following C codes , by removing the syntax and logical errors

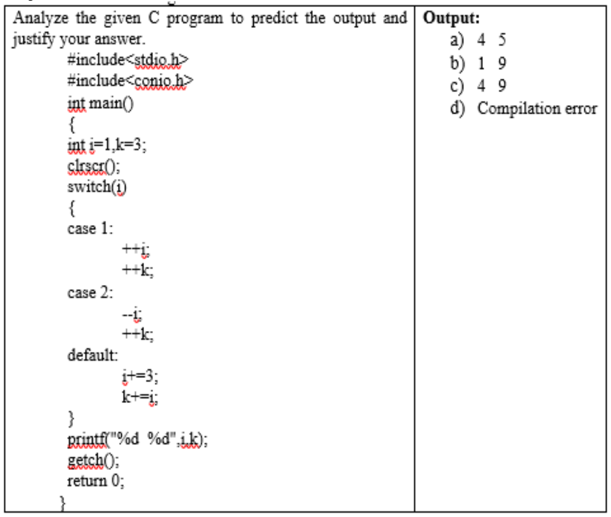
Program: To find maximum among three using relational and logical operator.

```
#include<stdio.h>
#include<conio.h>
int main();
{
int a,b;
clrscr();
printf("Enter three numbers");
scanf("%d%d%d",a,b,c);
if(a>b || a>c)
{
 printf("%d is maximum",a)
}
else if(b>a || b>c)
{
 printf("%d is maximum",b)
}
else
{
 printf("%d is maximum",c)
}
getch();
}
```

Fig. 8: Program analysis to detect the syntax and logical errors

10) Program analysis to predict the output of given code:

In this activity, the correct C code (without syntax and logical errors) was given to the students to predict the output. Students need to check the code line by line, need to analyze how the values of variables in the program go on changing while processing the code line by line. This is MCQ based activity, where 4 outputs were given to students as shown in fig. 9. They need to predict the correct output out of given. This kind of activity improves the analytical skill of students.



Analyze the given C program to predict the output and justify your answer.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i=1,k=3;
clrscr();
switch(i)
{
case 1:
        ++i;
        ++k;
case 2:
        --i;
        ++k;
default:
        i+=3;
        k+=i;
}
printf("%d  %d",i,k);
getch();
return 0;
}
```

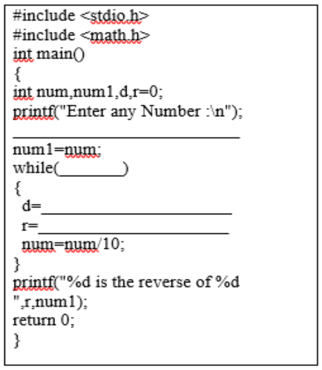
Output:

a) 4 5
b) 1 9
c) 4 9
d) Compilation error

Fig. 9: Program analysis to predict the output of given code

11) Half coding exercise:

The half-cooking code exercise also helps the students to improve their logic building, where the code with some missing lines was given to students. They need to complete the code as per the given problem statement. For example, complete the given C code by fill in the blanks with correct logic to find the reverse of the number as shown in fig. 10.



```
#include <stdio.h>
#include <math.h>
int main()
{
int num,num1,d,r=0;
printf("Enter any Number :\n");
_____________________________
num1=num;
while(_________)
{
  d=________________________
  r=________________________
  num=num/10;
}
printf("%d is the reverse of %d
",r,num1);
return 0;
}
```

Fig. 10: Half coding exercise

- ***<u>Active learning technique for Python Programming Course:</u>***

In addition to active teaching-learning techniques such as Predict the output, quizzes using online tools like Slido, Menti, Kahoot, in the Python lab course, students registered to Swayam portal. It is the initiative taken by the Govt. of India. On the Swayam portal, students opted course on “The Joy of computing using Python” which was offered as a MOOCs course. The course was designed for 12 weeks in which per week there were 6 to 7 videos along with the assignment. Students had to watch these videos and be asked to attempt the assignments as a part of their In Semester Evaluation (ISE).

At the end of the course, it was observed that students found satisfied with the way they learned the different concepts from the Python course. The performance of students was evaluated based on the scores obtained in all 12 assignments. It was a nice experience of learning for students.

All the active learning techniques that are discussed in this paper address the various challenges faced by students during learning the programming courses. Although some of these techniques are already proven it is necessary to correlate which technique is appropriate to overcome a particular challenge.

Challenges faced during implementation:

The use of these active learning techniques during the class requires much more time management.

The reluctance of students while using the visualization tools for programming courses.

VI. Assessment Methods for Programming Courses

The following methods will be used for assessment for Programming courses

- Continues Assessment for lab experiments
- Programming Challenges
- Programming test
- Debugging Test
- Technical Quiz
- Mini Projects
- Coding Assignments from platforms like HackerRank, Topcoder, HackerEarth

VII. Result and Discussion

We have applied the different active learning techniques described in this paper for programming courses like C, Python, etc in AY 2020-21. The In Semester Evaluation (ISE), Units Tests (UTs), and End Semester Evaluation (ESE) includes the questions from different categories which test students skill-set to write an algorithm, draw a flowchart, implement the program for a given statement, predict the output of given code, and analyses the code to remove syntax and logical errors that address the higher levels of the blooms taxonomy. This skill-set are very important for learning any programming language. The ISE, UT, and ESE examinations were conducted for 100 Marks.

For analysis of the result of implemented activities, we have compared the results of the C course taught at first-year level in AY 2020-21 and AY 2019-20.

Result clearly shows the performance of students has been increased after applying the different active learning techniques mentioned above for the programming courses.

A) A.Y 2019 – 20

In F.Y.B.Tech. 2019-20 (Sem-I), there are total 72 students are appeared for the C programming course in blended mode.

Elbow method: -

The unsupervised learning technique of machine learning is used to form and analyze the group of clusters as per marks of students. First, we applied the elbow method to find numbers of clusters as an input parameter required for K-Means.

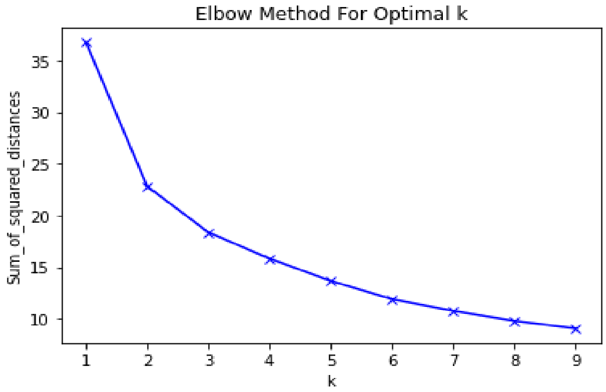


Fig. 11: Elbow Method to find k value for AY 2019-20 result data

As we can see from fig 11, the graph appears like an elbow. The plot looks like an arm with a clear elbow at k = 2. So, we can select k=2 as a number of clusters.

Silhouette method: -

We calculated the silhouette coefficients for the different values of k. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. The value near +1 indicates that the sample is far away from the neighboring clusters. A value near 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster.

TABLE II
Silhouette coefficient values

No. of clusters	Silhouette coefficient value
2	**0.5389387985608278**
3	0.5319637965991313
4	0.44513004025999436
5	0.4286151658768199

A higher Silhouette coefficient score relates to a model with better defined clusters. The value is higher k=2. So we can select the k=2 as the optimal number of clusters for given result data.

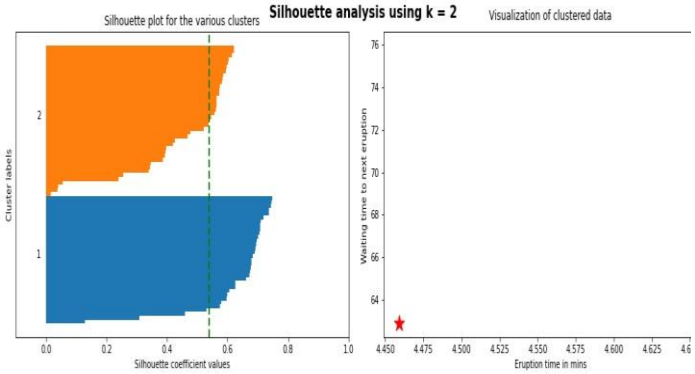


Fig. 12: Silhouette analysis for k=2

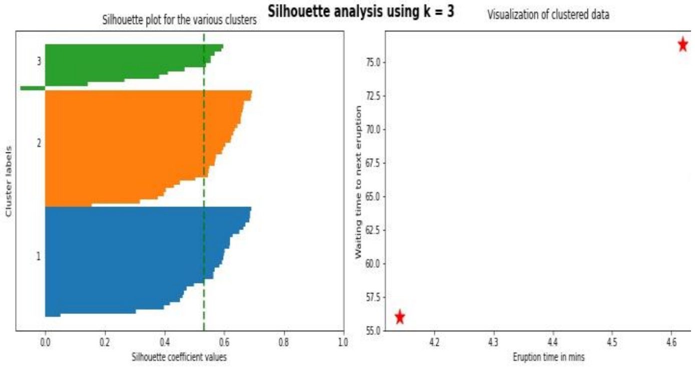


Fig. 13: Silhouette analysis for k=3

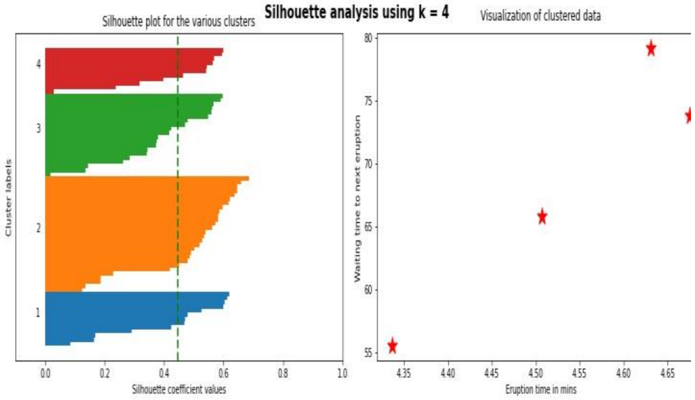


Fig. 14: Silhouette analysis for k=4

Fig. 12, 13, and 14 shows Silhouette analysis for k=2, 3, and 4 respectively for result data of AY 2019-20.

We have not selected k=3 and 4 due to wide variation in cluster size and student data is of less size. So k=2 will be the optimal number of cluster in this analysis.

From Fig. 12 it is observed that more students are in the lower-level cluster as compared to higher-level clusters which mean that more students are in the lower grades.

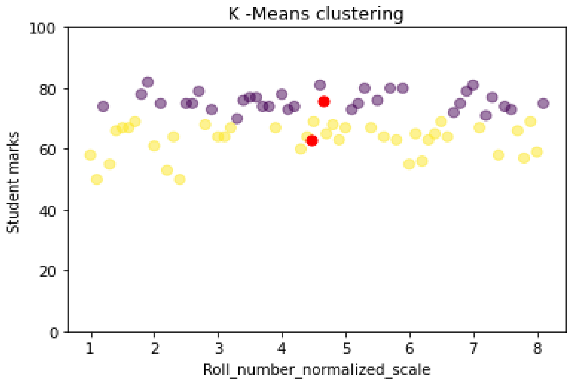


Fig. 15: Result of K-means for exam data set of AY 2019-20

The following table shows an analysis of the total number of students into two clusters, which indicate that the numbers of students in higher-level are less as compared to number of students in lower-level in AY 2019-20.

TABLE III
Size of the cluster for result in AY 2019-20

Cluster	Description	Size
1	Cluster 1(Higher level)	33
2	Cluster 2 (Lower level)	39

B) A.Y 2020 – 21

In F.Y. B.Tech. 2020-21(Sem-I), there are total 67 students are appeared for the C programming course in online mode.

Elbow method: -

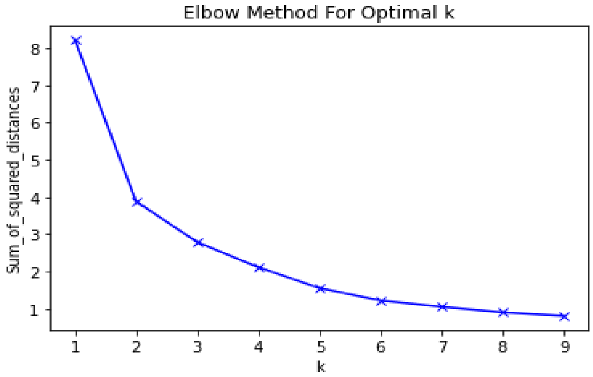


Fig. 16: Elbow Method to find k value for AY 2020-21 result data

As we can see from the fig. 16, the plot appears like an elbow. The plot looks like an arm with a clear elbow at k = 2. So, we can select k=2 as the optimal number of clusters for analysis.

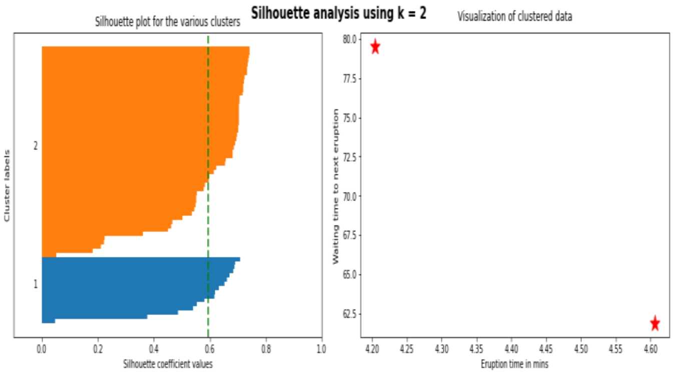


Fig. 17: Silhouette analysis for k=2

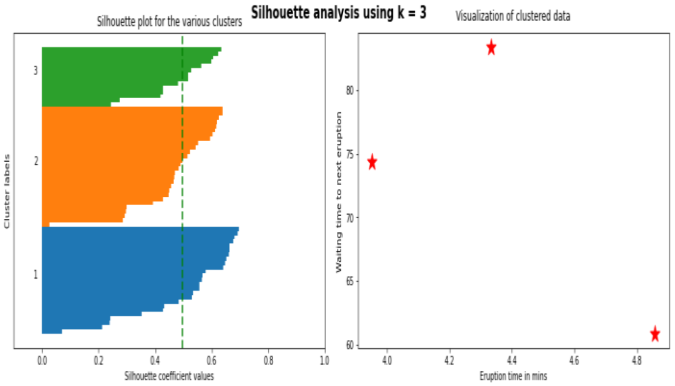


Fig. 18: Silhouette analysis for k=3

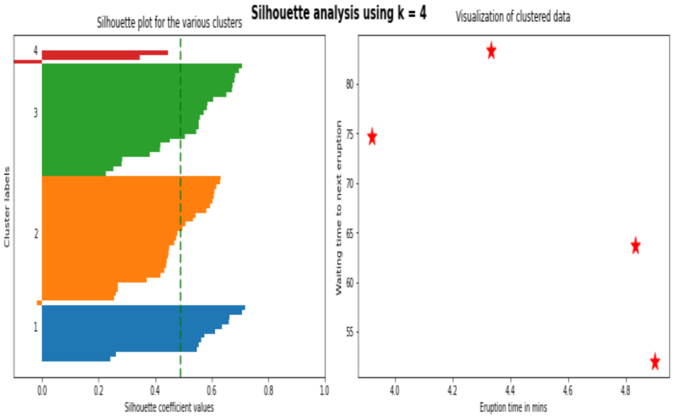


Fig. 19: Silhouette analysis for k=4

Fig. 17, 18, and 19 show Silhouette analysis for k=2, 3, and 4 respectively for result data of AY 2020-21.

From the fig. 17, it seems that there is no wide fluctuation in cluster size, and clusters are dense & well separated, so we have selected k=2 as the optimal number of clusters from silhouette analysis. As in Fig 17, from the thickness of the plot, the numbers of students in lower-level cluster 1 and higher-level cluster 2 can be visualized. The more numbers of students in the higher level cluster as compared to the lower level.

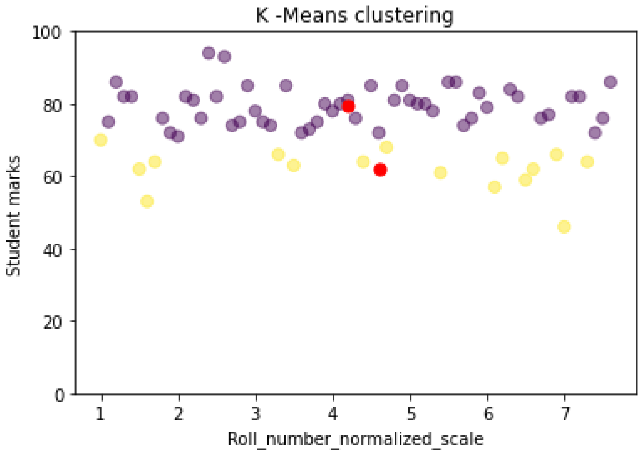


Fig. 20: Result of K-means for exam data set of AY 2020-21

Fig 20 shows the result of the K-means algorithm with input numbers of clusters k=2. The X-axis represents the roll numbers of students which are the normalized scale from our data and Y-axis represents the marks.

TABLE VI
Size of cluster for result in AY 2020-21

Cluster	Description	Size
1	Cluster 1(Higher level)	51
2	Cluster 2 (Lower level)	16

C) *Comparative Analysis of Results of AY 2019-20 and AY 2020-21*

The following table shows, the size of two clusters that is higher level and lower level. There are 72 students in AY 2019-20, out of which 33 are in the higher-level cluster and 39 are in lower-level cluster. And for AY 2020-21, there are total 67 students, out of which 51 are in higher-level cluster and 16 are in lower-level cluster.

From this cluster analysis, it has been observed that the students are moved to higher-level cluster in AY 2020-21 as compared to AY 2019-20.

TABLE V
Total number of students in clusters (C Programming)

Year	Size of higher-level cluster	Size of lower level cluster
2019-20	33	39
2020-21	51	16

From the centroid of each cluster, the arithmetic mean of marks of each student can be calculated. Table VI shows that the average marks of students in each cluster in AY 2020-21 have been increased.

TABLE VI
Average means of cluster

Year	Higher level cluster centroid position Y -Axis	Lower level cluster centroid position Y -Axis
2019-20	75.93	62.89
2020-21	79.52	61.87

TABLE VII
The grade comparison

Grade	AY 2019-20	AY 2020-21
AA	08	10
AB	18	28
BB	07	14
BC	27	10
CC	09	03
CD	03	01
DD	00	01
FF	00	00

From the grade comparison as shown in table VII, the number of students shifted from lower grade to higher grade in A.Y 2020 -21.

TABLE VIII
CO attainment comparison

Sr. No.	CO	CO Statement	Threshold based Attainment %	
			AY 19-20	AY 20-21
1	CO1	Explain the basic terminology and concepts of c programming language.	75.58 %	89.83 %
2	CO2	Construct the algorithm and draw a flow chart to solve the problem.	64.63 %	87.73 %
3	CO3	Write a c programs to solve given problems	78.81 %	89.32 %
4	CO4	Examine the given c program to remove the logical & syntax errors and predict the correct output	67.34 %	88.64 %

Table VIII shows the CO attainment comparison of the C programming course of the AY 2019-20 and AY 2020-21. The result of CO attainment shows that the required skills for programming like designing algorithms, drawing flowcharts, writing code, debugging of the program to predict the logical & syntax errors and to predict the correct output got improved.

As like result analysis of the C programming course, the cluster validation techniques i.e Elbow and Silhouette are also used for the result analysis of the Python programming course taught at the second year level in AY 2019-20 and AY 2020-21. Table IX shows the summary obtained after cluster analysis of results.

From the analysis of the result, it shows that there are more numbers of students are shifted to higher-level cluster in AY 2020-21 as compared to AY 2019-20.

TABLE IX
Total number of students in clusters (Python Programming)

Year	Size of higher-level cluster	Size of lower level cluster
2019-20	47	15
2020-21	64	19

The use of several active learning techniques and visualization tools that are discussed in this paper is really helpful to students for active participation during teaching-learning of programming courses and make them understand programming courses effectively.

CONCLUSION

The IT industry demands graduates with sound technical competency in the programming domain. Therefore, the effective delivery of programming courses is very important which plays a vital role in creating interest among the students to learn the programming languages. In this paper, we have discussed the different tools and active learning techniques that were implemented for programming courses such as C, Python, etc. By incorporating the various visualization tools and active learning approaches, we can increase the active participation of a student in programing courses which reduce the fear of programming and enhance logical thinking, problem-solving ability, debugging, and analysis skills of students. The different evaluation methods we have applied for assessment of programming course such as continuous assessment sheet, programming challenges, debugging test, technical quiz. The result of implementation shows that the performance of the students has been increased. The number of students are shifted from lower grade to higher grade as well as the CO attainment got increased. The use of visualization tools and activity-based learning for programming courses is very essential for inculcating the required programming skills among the students

REFERENCES

[1] Settle, A., Vihavainen, A., & Miller, C. S. (2014). Research directions for teaching programming online. In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[2] Srinivasan, S., &Centea, D. (2018, October). An active learning strategy for programming courses. In Interactive Mobile Communication, Technologies and Learning (pp. 327-336). Springer, Cham.

[3] Derus, S. R. M., & Ali, A. Z. M. (2014). Integration of visualization techniques and active learning strategy in learning computer programming: a proposed framework. International Journal on New Trends in Education and Their Implications, 5(1), 93-103.

[4] Martínez, C., & Muñoz, M. (2014). ADPT: An active learning method for a programming lab course. In Proceedings of the 10th International CDIO Conference.

[5] Gaspar, A., &Langevin, S. (2007). Active learning in introductory programming courses through student-led “live coding” and test-driven pair programming. In International Conference on Education and Information Systems, Technologies and Applications, Orlando, FL.

[6] Duffany, J. L. (2015, July). Active learning applied to introductory programming. In XIII Latin American and Caribbean Conference for Engineering and Technology.

[7] Jaithavil, D., &Kuptasthien, N. AN ADAPTIVE ALGORITHM FOR LEARNING COMPUTER PROGRAMMING COURSE. In The 15th International CDIO Conference.

[8]Gorvankolla, A. K., Navya, R., &Smitha, G. R. Adopting gile Technology in Learning C Programming.

[9] Liu, S., Chen, Z., & Tang, J. (2013, June). The Improved Methods of Teaching Practice Based on C Language Programming. In International Conference on Education Technology and Management Science (ICETMS 2013) (pp. 1361-1364).

[10] Lathigara, A., Tanna, P., & Bhatt, N. (2021). Activity Based Programming Learning. Journal of Engineering Education Transformations, 34, 499-506

[11] Thorat, S. A., &Kshirsagar, D. P. (2021). Developing Logic Building, Problem Solving, and Debugging Programming Skills Among Students. Journal of Engineering Education Transformations, 34, 402-406.

[12] Reddy, P. C. (2015). Analysis of teaching computer programming in Indian Context. Journal of Engineering Education Transformations, 28(4), 50-55

[13] Doddamani, S. T. (2018). Project Based Learning of Programming Subject: Case study on Data Structures. Journal of Engineering Education Transformations, 31(3), 250-255.

[14] Liu, S., Chen, Z., & Tang, J. (2013, June). The Improved Methods of Teaching Practice Based on C Language Programming. In 2013 Conference on Education Technology and Management Science (ICETMS 2013) (pp. 807-810). Atlantis Press.

[15] Handur, V. S., Kalwad, P. D., Yaligar, N., Garagad, V. G., &Pawar, M. K. (2015, October). An activity based learning: C programming. In 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE) (pp. 310-314). IEEE.

[16] Mouenis Anouar Tadlaoui1 and Mohammed Chekour2, A blended learning approach for teaching python programming language: towards a post pandemic pedagogy, International Journal of Advanced Computer Research, Vol 11(52) ISSN (Print): 2249-7277 ISSN (Online): 2277-7970 http://dx.doi.org/10.19101/IJACR.2020.1048120.

[17] Namratha M , Rekha G.S , Syed Akram 1 2 3, Selva Kumar S4, Jyothi S Nayak5, Active Learning Approach for Python Programming, Journal of Engineering Education Transformations ,volume 32 , No. 1, July 2018, ISSN 2349-2473, eISSN 2394-1707.

[18] Geetha Kiran Annegowda 1, Mohana Lakshmi J2, Student Centric Pragmatic Approach to Impart Concepts of Python Applications Programming, Journal of Engineering Education Transformations ,Volume 34 , No. 1, July 2020, ISSN 2349-2473, eISSN 2394-1707.

[19] https://www.polleverywhere.com

[20] https://kahoot.com

[21]https://raptor.martincarlisle.com/

[22] http://pythontutor.com/