Journal of Engineering Education Transformations,
Volume No 36, January 2023, Special issue, eISSN 2394-1707

10.16920/jeet/2023/v36is2/23090

Task Based Assessment: An Innovative Methodology for Studying and Assessing High Level Programming-Oriented Courses

Paresh Tanna, Amit Lathigara, Nirav Bhatt
School of Engineering, RK University, Gujarat, India

Abstract: **Many students are being struggled with high level programming related courses. It is an enthusiastic and dynamic approach to teach and assess such courses for the development of their programming skills. Also, traditional approach to evaluate such high-level programming-oriented courses is with list of programs repeatedly being followed in consecutive batches as well as theory exam is conducted that generally focuses on remembering/understanding oriented questions. During the study tenure, students get very less opportunities to write or solve technical problems or tasks as per the IT industry needs as well as they get struggle to be selected in a reputed IT company by clearing the technical examination through the campus drive. Also, IT companies do not follow only the mechanism i.e., "writing code from scratch for all projects". In fact, many more situations arise where students must outperform similarly. Such circumstances can be discovered depending on the work culture, constraints, coding elegance, etc. within each industry. Proposed method is one of such mechanisms out of these i.e., each problem is being divided into different tasks. Each task may represent a requirement of client or a set of small requirements. Here, the proposed innovative methodology is "Task Based Assessment (TBA)" for high level programming-oriented courses. This would focus on outcome based learning and innovative assessment approaches. The results obtained with the conventional approach are compared to the results obtained with the TBA approach, and it is discovered that a greater number of students obtained more than B+ grade (>=70%) with the proposed TBA approach, as also statistically proven. This correlation indicates that students are more conscious and interested in the TBA approach as opposed to mugged up & writing hypothetical ideas into the examinations. Furthermore, student feedback has been found to be extremely positive. Also, this would help students to improve their programming skills to cope with the IT industry challenges.**

***Keywords*: High Level Programming Languages, Programming Skills, Technical Skills, IT Industry Challenges, Task Based Assessment, Innovative Assessment**

***JEET Category*—Practice**

I. INTRODUCTION

MANY students nowadays choose computer engineering or information technology as a professional path. As a result, institutions and colleges are getting student masks with varying talents and from various regions. Many of them are lacking with technical/practical experiences since they have memorized a lot of theory in order to get excellent grades during their schooling, but they have unknowingly missed a lot that may benefit them with engineering studies. For many students, the nature of learning was established solely by mugged up. The need for creative engineers is both worldwide and timely. Amongst issues confronting society, the need for innovative problem solvers is becoming increasingly apparent. The National Academies and political leaders have urged forward-thinking engineers to drive the engine of a country (National Governors Association, 2007; Perry et al., 2008). At a higher level, engineering educators and administrators might consider innovation in curriculum creation, teaching and assessment (Brent and Felder, 2014). (Gilbertson, 2014) stated that confident gained via 'practical exposure' were a better road to a happy future for students than grades and mugging up. Task-based learning has gained prominence in engineering students' teaching because it allows students to utilize programming language as a medium of career set up in the IT field rather than focusing on clearing and high scoring in a particular course. In other words, task-based training develops students' capacities to attain technical skills development goals in real-life circumstances in addition to conceptual knowledge build-up. Furthermore, (Skehan, 1998) believes that task-based learning would help to grow and improve the logical and naturalistic acquisitional mechanisms. In reality, logical and technical abilities are extremely important for computer engineering students. These in-demand competencies may be

obtained through programming language courses (Sharma, A.,

2019). As a result, it is more vital for faculty to instruct and evaluate logical/technical abilities based on the requirements of concerned programming language courses in the IT sector. (Senthil, 2020) stated to offer young students with the necessary information and skills throughout engineering in the specified course of engineering for improved learning and career prospects.

II. LITERATURE REVIEW

Adnan et al discovered that innovative methods for teaching programming can aid in the development of learners' programming concepts by improving their cognitive ability to develop a mental model, increasing their engagement, and stimulating their abstract thinking in cognitive development (Adnan, 2022).

Omeh et all stated that teaching computer programming with innovative pedagogy results in significantly increased student academic achievement in programming skills, digital skill development, and self-efficacy in the experimental group than in the control group (Omeh, 2022).

The traditional approach to assessing student comprehension and abilities includes a combination of programming assignments and written tests. Teachers frequently assign out-of-classroom assignments designed to assess student program writing abilities, and they administer in-class tests to assess how well students understand the concepts. These tests are taken and used as an additional check on language syntax and logical structure (Smith, P. P., 2005). While proposed approach is innovative and reflective at every point from lab assignment submission to examination.

Conventional methodology examines and tests the source code without running the programme (Ala-Mutka, 2005) and is used to evaluate programming style, linguistic structure and semantic error, programming metric examination, basic closeness examination, keyword detector, plagiarised content finding, and outline analysis. This is indeed depending on the manual evaluation of knowledge and also limited to whole program submission only. It does not help for higher level programming.

The traditional evaluation approach, in which a single written assessment counts toward a student's final grade, would never again meet the new requirements of programming language courses (Wang, Y., 2012).

Thoman et al specified that over the last few decades, a large number of task-based programming environments have been developed, and the task-based parallelism paradigm has proven to be widely applicable for client applications (Thoman, 2018).

Marco et al explored the use of tasks with the way navigation and task modelling can be used to help with web application design and also how these two models could be used in tandem to improve the development of web applications (Marco, 2002).

Djambong et al stated that an inventive assessment was required as a consequent teaching perspective to establish stronger empirical evidence of possible relationships between students learning and placement (Djambong. 2016).

De Ruiter et al mentioned that the correlation between CSA (Coding Stages Assessment i.e., proposed by them) scores and computational thinking ability demonstrated structure rationality. The items have high discrimination levels and a range of difficulty levels to capture different levels of proficiency (de Ruiter, 2021).

Douce et al., in 2005, created Automated System for Assessment of Programming (ASAP) code which serves as an assessment tool in the e-learning structure.

Hwang et all demonstrated a Two-Tier Test-Based Programming Training (T3PT) approach that it was significantly superior to the conventional programming learning approach in terms of students' programming logic concepts, problem-solving awareness, technology acceptance, and learning approach satisfaction (Hwang, 2022).

Learning through engagement activities inspire the 'learning' process and develop soft skills, logical skills, practical skills, and other skills that lead to the notion of an engineering education that attempts to narrow down the gap between theory concepts and actual performance in programming courses in higher semesters of computer engineering (Tanna, 2022).

Nagappan et al explored the programmers with pair programming were increasingly independent, performed better on tasks and tests, and were guaranteed to get higher skills than their independent partners. But this is limited to very basic level programming courses (Nagappan, 2003).

The teaching-learning process is the responsibility of the teacher in the classroom. Interaction between students for programming courses occurs when they encounter programming problems such as debugging errors (Rienovita, 2014).
IT professionals and Academicians place a vital position on teaching and learning programming languages. Students should be able to comprehend any random assignment using any of the programming languages. More importantly, students should be attentive in improving application development performance by utilising fundamental and advanced programming language concepts (Husain, M., 2015).

The few of methodologies used in reviewed mechanisms is bounded to a whole program, and some are only restricted to a particular programming language. For evaluating student

programming skills, an increasingly productive all-adopting methodology or approach is required. However apart from the recognized barriers, the program quality and semantic comparability with many parameters should be improved for the development of programming or technical skills. Based on this, the proposed approach i.e., TBA is intended for all types of programming languages with varying task formats.

III. METHODOLOGY

"Education is not the learning of facts, but the training of the mind to think." ~ Albert Einstein

NEP 2020 guides us about "holistic and multidisciplinary education, optimal learning environments and support for students, motivate, energized, and capable faculty etc." and further helps to deal with the challenges of education system.

"We are teaching. Are they learning?" stated by IUCEE. "Is this taking place among your class/students?". To be successful with this, proposed assessment technique i.e., TBA would focus more on programming skills development.

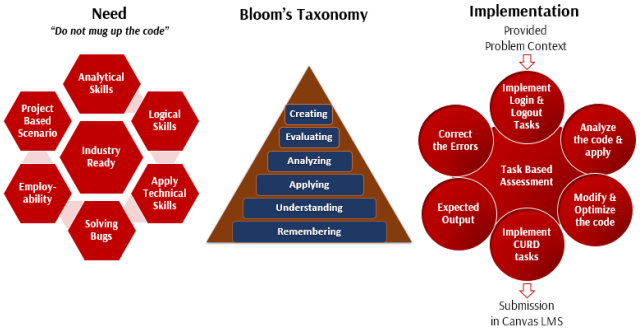

Fig. 1. TBA Approach Alignment

Conventional pen paper-based approach focusses more on mugged up method i.e., answering the questions based on students' ability to remember or understand. This does not evaluate students according to the required programming skills in the IT sector. Also, weightage of the theory portion in such courses is much higher so students also redirect them towards more mugged up concepts to score maximum marks rather focusing on programming skills development. To deal with such challenges in conventional methods and improving the programming skills, one of the important methodologies is assessing the required skills with Task Based Assessment (TBA).

Fig. 2 highlights the CLOs for some high-level programming-oriented courses. By considering these CLOs, traditional pen and paper examination may not assess the students' programming skills to fulfill such type of CLOs. For example, one of the CLOs is "design mobile application using various UI components and layouts", to fulfill this CLO pen and paper examination may not be the proper way rather it should be directly assessed through computer i.e., actually being followed with TBA approach. Similarly, all such highlighted CLOs can be achieved through TBA approach under the faculty guidance and student learning approach.

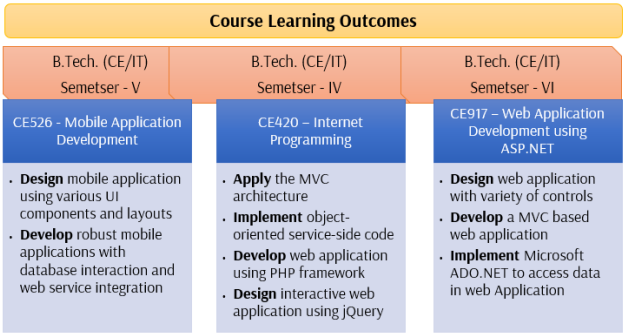

Fig. 2. Course Learning Outcomes for Some High-Level Programming Oriented Courses

TBA methodology works on assessment of the student's ability to analyse, evaluate, and apply the technical knowledge of high-level programming-oriented courses. This would improve the students' competence and attention to perform the desired task solution. Continuous evaluation plays a vital role for engineering students, and this proposed methodology entirely emphasis on the continuous evaluation i.e., as shown in Fig. 3, lab experiments are assigned based on TBA approach through which depth understanding about methodology and real time practices guide students for the upcoming examinations followed by doubt solving sessions. TBA approach was proposed for all three types of university theory examinations i.e., TCIE-I & II (Theory Continuous Internal Examination) and TSEE (Theory Semester End Examination). Before conducting these examinations, mock examination was conducted followed by student feedback and result analysis. This gave students much confidence about the university upcoming examinations by solving relevant doubts. With the same pattern i.e., followed with mock examination, both the internal examinations (TCIE-I & II) was conducted followed by result and feedback analysis. Distinguishing and sharing mistakes found in these examinations, may help students about a great deal for development in programming skills.

Execution of TBA Approach:

Following steps were followed for the execution as referring Fig. 1:

- Canvas platform was used for exam in the computer lab.
- A question paper was formed with different type of tasks from diverse clusters like login and logout tasks, analyse the code and apply to solve the assigned tasks, modify/optimize the existing code to get the desired outcome, implement CURD operation tasks, write the code to get the desired output (as discussed in sample questions), few tasks related to correcting the errors like logical errors, syntax errors etc.
- For each task separate instructions along with screenshots was provided to write the solution for the same.
- For TCIE-I & II, total 8 tasks which would be of 2 marks and 4 marks were provided to be solved within 90 minutes of duration. For TSEE, total 20 tasks which would be of 2 marks and 4 marks were provided to be solved within 180 minutes of duration.

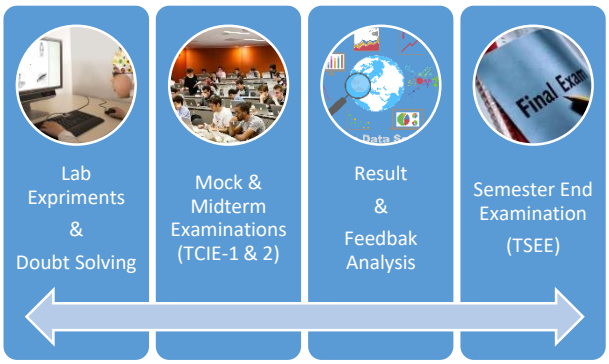

Fig. 3. Execution Steps for TBA Approach

- After submission, faculty has done the assessment for each submission by following rubrics which was already shared with students at the beginning of semester.

Fig. 4. Assessment Design for TBA Approach

Assessment Design for TBA Approach

Each task would be based on industry-oriented skills set with reference to Fig. 1 and students apply their knowledge to solve the given tasks as well as course faculty assessed each task submitted solution using the rubrics highlighted in this paper in Table 1. Assessment design for TBA approach is highlighted in Fig. 4 followed by description.

A. Importance (Why):

TBA is significant which would work as a beacon for students to be industry ready by improving their analytical skills & logical skills, apply technical skills, solving bugs, work with project-based scenario, employable after completing study etc.

B. Purpose (What):

The purpose of TBA approach is to make students well equipped with industry ready skills set by continuous evaluation during lab hours with the help of lab assignments, continuous midterm internal examinations, and final term end examination.

C. Schedule (When):

To assess learning continuously, lab experiments are assessed abidingly as well as examinations were conducted during its university regular schedule.

D. Location (Where):

The examination conducted in computer laboratory through canvas platform with seating arrangement in a unique pattern i.e., each student get unique tasks list among nearby students.

E. Benefits:

Students benefited with TBA approach compared to traditional pen and paper approach i.e., i) practiced with self-learning strategy to develop their programming skills, ii) real time problem understanding and finding the solutions iii) improvement in technical knowledge for high-level programming-oriented courses rather than writing imaginary conventional hypothesis answers.

F. Challenges:

Moving from traditional to innovative TBA approach, faculty may find some noticeable challenges also: i) submission pattern from students was different which took some additional time for assessment, ii) wrote large number of lines for simple solution, iii) design of multiple question papers with unique list of tasks i.e., through own testing (compiling and executing) of each task in the beginning takes higher time, iv) creating awareness about the TBA approach throughout the semester.

G. Sample Tasks for TBA Approach:

Students are spending too much time during their classroom learning but faculties are not fully aware about the outcome of teaching. In fact, the way students grabbed the concept may also create a worry for the faculty i.e., many students are mugging up the concepts to score maximum marks or to score passing marks. This may not solve the purpose of engineering education. Some sample tasks are discussed here with bloom's level alignment for different type of programming-oriented courses. Level of tasks can be much higher based on the course content and weightage into the course. Here, we have mentioned few tasks that are being completed during the examination with limited stipulated duration. Expectations may be higher for tasks but accordingly examination duration should be allocated much higher to get the relevant and comparatively improved outcome.

Task-1: Modify the assigned code to display Issue-Date, Due-Date and Return-Date in proper format as per the given output. Also perform edit and delete operations for user management.

This type of task would check the analytical skills and redirect the students to apply their practical skills.

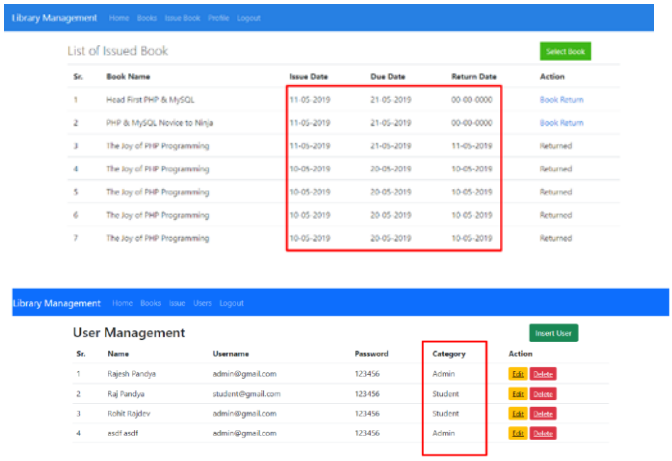

Fig. 5. Sample Task – 1 Screenshot

Task-2: Implement the "Delete" functionality along with confirmation message before deleting the data.

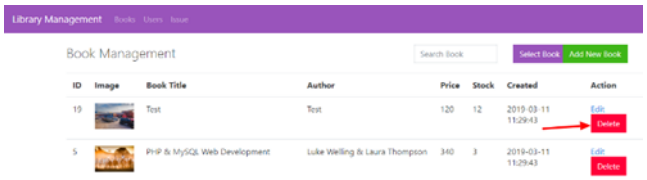

Fig. 6. Sample Task – 2 Screenshot

Also, student programming skills related to project development can be evaluated with such type of tasks and assist the faculty to explore whether the student has studied the subject matter correctly or not.

Task-3: Implement code to display total number of books in the library and total number of books issued.

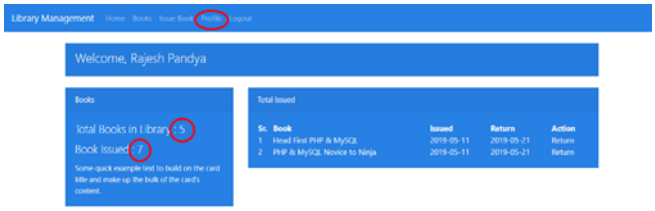

Fig. 7. Sample Task – 3 Screenshot

This type of tasks would help faculty to check whether students are able to fetch the values from the database and able to display fetched data in correct format. Also, faculty can revise the topics based on results and feedback from students.

Task-4: Write code to insert book details in the books table.

This type of tasks would habituate the student to do such operations i.e., the primary requirement of any real time project in industry.

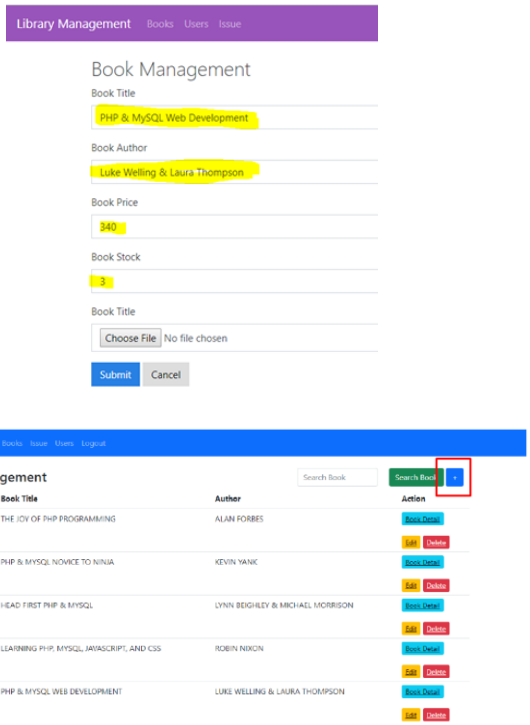

Fig. 8. Sample Task – 4 Screenshot

Tasks 5 to 7 are highlighted here to demonstrate real time mobile application development. Such of type of tasks keep students engaged with the concepts' learning either in a group or individually.

Task-5: Write code to display registered users in a list view as per above output. Or Write code to display selected book details.

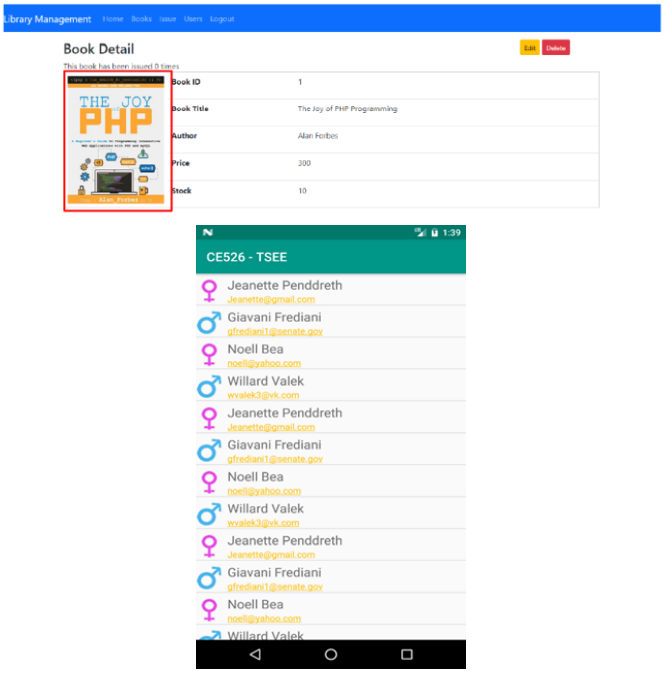

Fig. 9. Sample Task – 5 Screenshot

Task-6: Develop a functionality to display current battery level.

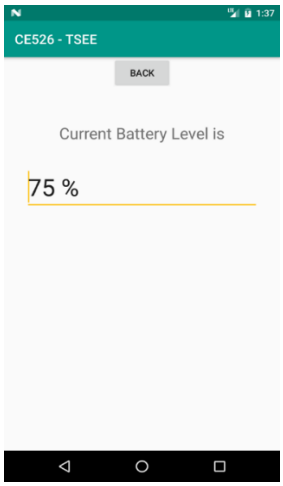

Fig. 10. Sample Task – 6 Screenshot

Task-7: Write the missing code to complete the registration functionality.

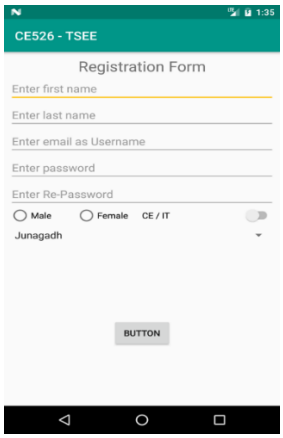

Fig. 11. Sample Task – 7 Screenshot

Task-8: Develop new module(s) to perform insert, update and delete operations on database table.

Such type of tasks prepares students to write code and develop module from scratch which help them for project development as well as to understand the existing work of already developed projects.

Remembering to creating level of Bloom's taxonomy, have been covered with TBA approach in contrast to traditional pen and paper examination. This helped to increase the programming skills as well as self-assurance of students for real life project development. TBA has also provision to write comments for each task solution that helped students to be habituated with a regular practice being followed in industry.

Tasks Format Alignment with Blooms Level (Dontham, 2016)

A. Remembering level: Task-1 is normally being asked in conventional way as: Explain the mechanism to fetch data within webpage.

B. Understanding level: Taks-2 is normally being asked in conventional way as: Write steps to delete a record from table.

C. Apply level: Tasks 1,2 and 3 are samples of this level i.e., rather than mug up with the concepts, students would apply their knowledge and would promote students for higher level thinking to solve the problem that can be employed to be evaluated by TBA approach.

D. Analyze level: Task 1 & 4 indicates an analytical level, where students need to provide solution with the given specifications by comparing the functionality available in the other applications also.

E. Evaluate level: Task 5 & 7 highlights the evaluation level specifications that demands an innovative approach from students to solve it with implications of all types of testing.

F. Create level: Task 6 & 8, states to the maximum intellectual direction where students are inquired to develop functionality from scratch that redirects students to think at bloom's creation level to provide the solution using acquired programming skills.

Such type of sample tasks helps to achieve and assess 360-degree perspective of programming skills of higher-level programming-oriented courses.

Rubric:

TABLE I
RUBRIC FOR TBA APPROACH

	Excellent	Good	Average	Poor
2 - Marks Task	**2**	**1.5**	**1**	**0.5-0**
4 - Marks Task	**4**	**3**	**2**	**1-0**
Implementation with proper sequence, syntax, and comments	Student code shows complete (100%) implementation of the given task. All steps are in correct sequence. Code implementation is without any errors. All comments are put up properly.	Student code shows complete/partial (75%-100%) implementation of the given task. All steps are in correct sequence, or few steps are not in correct sequence. Comments are not put up properly.	Student code shows more than 50% implementation of the given task. With few errors and comments are not put up properly.	Student code shows more than 20% implementation of the given task. With many errors. Comments are not put up properly.

Outcomes:

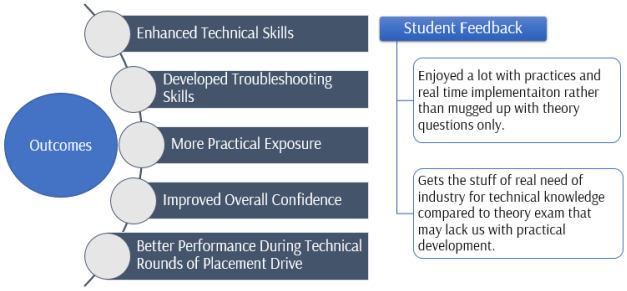

Fig. 12. Expected Outcomes for TBA Approach

Fig. 12 highlights the expected outcomes from the proposed TBA approach along with some sample feedback from student.

Student Guidelines:

- This component evaluation will be conducted during regular examination schedule.
- Each student will get a set of requirements into the form of tasks.
- Assigned tasks will contain different types of questions which will be of 2 marks and 4 marks.
- Read the problem description, requirements & other details carefully and implement for the solution.
- This assessment is intended to test the hands-on skills related to different topics in the course. Students will apply the different skills to accomplish the assigned task.
- Students will be provided with Supplied Files folder that contains project files and expected UI screens. Download and extract the Supplied Files folder into local drive and name folder as your "Enrolment No.".
- Modify project files as per the task description provided to you in the question paper.
- Once all the tasks are implemented, kindly prepare the "EnrollmentNo.zip" file of your project along with database supported files also.
- The completed work should be submitted through Canvas LMS.

IV. Result Analysis

The impact of TBA approach is examined with "Web Application Development using ASP.NET" course i.e., one of the high-level programming-oriented courses and the result of B.Tech. (CE and IT branches) sixth semester students (A.Y. 2020-21 and A.Y. 2021-22) are firmed in the Table 2 with key factors (Ganesh, 2018).

TABLE II
RESULT ANALYSIS FOR WAD USING ASP.NET COURSE

Result Analysis (For AY 2020-21 & 2021-22)			
Level	**Grade**	**% Students with Conventional Pen and Paper Approach (AY 2020-21)**	**% Students with TBA Approach (AY 2021-22)**
Outstanding	A+	2.63	9.52
Excellent	A	10.53	47.62
Very Good	B+	15.79	19.05
Good	B	26.32	14.29
Above Average	C+	18.42	4.76
Average	C	7.89	4.76
Poor	D	7.89	0.00
Fail	F	10.53	0.00

Result of A.Y. 2020-21 demonstrates that around 29% students who had scored equal to or higher than over B+ grade with conventional pen and paper method while result of A.Y. 2021-22 which was implemented with TBA approach highlights that around 76% students had scored equal to or higher than over B+ grade and this comparison can be referred in Fig. 14.

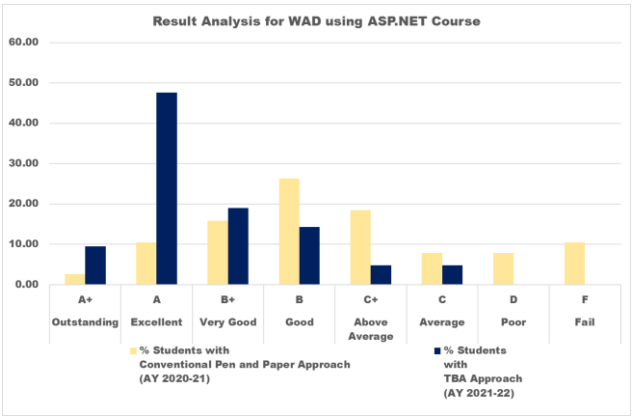

Fig. 13. Result Analysis Comparison

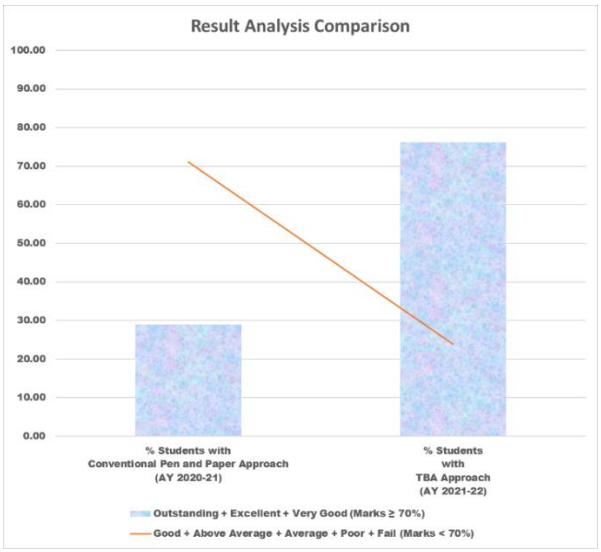

Fig. 14. Result Analysis Comparison between ">=70% Students Group" & "<70% Students Group"

This assessment demonstrates that students are more evident and fascinating about TBA approach instead of mugging up theories and writing the answers with conventional pen and paper approach.

As shown in Fig. 13, conventional pen and paper method is being conquered with TBA approach i.e., significantly result improvement found for the proposed approach.

Fig. 14 shows that the higher number of students with grades (>=70%) for TBA approach found compared to the conventional approach.

The Control group and experimental group are set up to evaluate the effect of the proposed approach on student's knowledge. The control group i.e., Conventional approach (A.Y. 2020-21 grades i.e., given in last second column of Table 2) and experimental group (AY 2021-22 grades i.e., given in last column of Table 2) i.e., TBA approach is matched in

contrast to each other in this experiment. The factual analysis was applied to exemplify the mean and the standard deviation of the score. Since the secured grades for conventional and TBA approach were composed for the same course in two different years, the paired t-test was applied to assess the substantial difference for both exams' grades. The paired t-test was performed to test the hypothesis.

Table 3 shows the collective tests' results of conventional and TBA approaches. The overall finding displays that there is more than 2 times enhancement with the TBA approach for grades (>=70%).

TABLE III
THE PAIRED SAMPLES STATISTICS OF CONVENTIONAL AND TBA APPROACH

Level	Conventional Approach (%)		TBA Approach (%)	
	Mean	Standard Deviation	Mean	Standard Deviation
Grades >=70%	9.65	6.62	25.40	19.83
Grades <70%	14.21	8.03	4.76	5.83

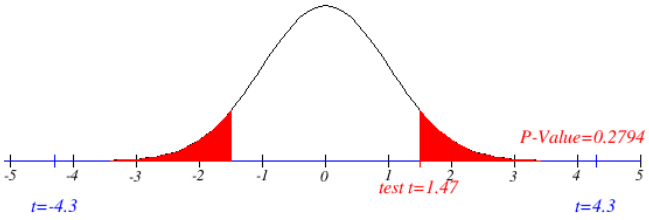

Fig. 15. Results of Paired Samples t-Test over Grades (>=70%)

TABLE IV
RESULTS OF PAIRED SAMPLES T-TEST OF TRADITIONAL AND PBL APPROACH

Level	Paired Differences (%)		t-test result
	Mean	Standard Deviation	(2-tailed)
Grades >=60%	15.75	13.21	0.27

The hypothesis testing with paired sample t-test was utilized for this study. The results are anticipated to suggest higher mean value for TBA approach than conventional approach.

The null hypothesis H0: There's no distinction in mean for the TBA approach than the conventional approach.

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

Table 4 and Fig. 15 show the findings of paired t-test for grades (>=70%). At 27% significance level, all null hypotheses are rejected and t-test result i.e., P-value indicates that with TBA approach results has been improved undoubtedly compared to conventional approach. In such a way, it can be concluded that there is a statistically remarkable contrast between mean score of the conventional approach and TBA approach. Subsequently, it is noticeable that the TBA approach can upgrade the students' programming skills essentially.

As discussed earlier, level of tasks can be much higher based on the course content and weightage into the course. Here, we have compared the given tasks with the traditional pen paper approach which is totally mugged up examination. Result improvement not only suggests the higher grades but also shows the way towards skills development for programming-oriented courses i.e., the need of current era in the IT field.

Post Implementation

Student Feedback:

For the proposed approach, the feedback was conducted from students' groups and around 86% students shown positiveness towards TBA approach. This feedback shows the students' understanding of the concepts and enthusiasm for the TBA approach towards WAD using ASP.NET course.

Facilitator Reflection:

For the TBA approach, the reflection from facilitator was conducted and found more positive towards TBA approach. This feedback indicates the favorableness towards TBA approach for WAD using ASP.NET course.

V. CONCLUSIONS

Lab assignments and examinations based on TBA approach may assist learners to learn high-level programming languages in an interactive learning environment. Besides this, the IT sector seeks applicants who would be talented and have also adequate knowledge of tasks-based problem-solving skills along with practical knowledge of languages. Lots of practises with TBA pattern may help students for the industry work i.e., to write code or provide solution from scratch or edit code written by peers within the same or different groups. Learning and assessment based on such a pattern motivated students to perform much better with the high-level programming-oriented courses.

REFERENCES

Skehan, P. (1998). *A cognitive approach to language learning*. Oxford University Press.

Winckler, M., Palanque, P., Farenc, C., & Pimenta, M. S. (2002). Task-based assessment of web navigation design. *CEP*, *91501*, 970.

Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. *ACM Sigcse Bulletin*, *35*(1), 359-362.

Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer science education*, *15*(2), 83-102.

Smith, P. P. (2005). Techniques of assessment pertinent to computer programming courses. In *38th Midwest Instruction and Computing Symposium, University of Wisconsin-Eau Claire, WI*.

Douce, C., Livingstone, D., Grindle, S., Cobb, J., & Orwell, J. (2005). A technical perspective on ASAP-Automated

system for assessment of programming. In *Loughborough University*.

Perry, W., Broers, A., El-Baz, F., Harris, W., Healy, B., & Hillis, W. D. (2008). Grand challenges for engineering. *National Academy of Engineering, Washington, DC*, *83*.

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.

Rienovita, E., & Takeuchi, Y. (2014). Encouragement of skill mastering motive based on need for approval and Interaction. *Human Interface*, 235-238.

Brent, R., & Felder, R. (2014). Want your students to think creatively and critically? How about teaching them?. *Chemical Engineering Education*, *48*(2), 113-114.

Gilbertson, A. (2014). 'Mugging up'versus 'exposure': international schools and social mobility in Hyderabad, India. *Ethnography and Education*, *9*(2), 210-223.

Husain, M., Akki, M. S., Maralappanavar, M. S., & Narayan, D. G. (2015). Designing and Teaching of Language Independent Object Oriented Programming Course. *Journal Of Engineering Education Transformations*, 248-251.

Djambong, T., & Freiman, V. (2016). Task-Based Assessment of Students' Computational Thinking Skills Developed through Visual Programming or Tangible Coding Environments. *International Association for Development of the Information Society*.

Dontham, R., Patil, S. P., Patil, M. A., & Ingale, A. K. (2016). Educating IT-Engineering Theory Courses Using BLOOMS Taxonomy (TC-BT) as a Tool and Practical Courses Using Magic of Making Mistakes (PC-MoMM): An Innovative Pedagogy. *Journal of Engineering Education Transformations*, *29*(3), 75-84.

Thoman, P., Dichev, K., Heller, T., Iakymchuk, R., Aguilar, X., Hasanov, K., ... & Nikolopoulos, D. S. (2018). A taxonomy of task-based parallel programming technologies for high-performance computing. *The Journal of Supercomputing*, *74*(4), 1422-1434.

Ganesh, K. E., & Pranesha, 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.

Arora, A. S. A. S. (2019). Active learning: an Instructional Technique for Improving Educational Practices. *Journal of Engineering Education Transformations*, *32*(3).

Senthil, R. (2020). Enhancement of Engineering Education by Incorporating Active Learning Methodologies. *Journal of Engineering Education Transformations*, *34*(1), 2349-2473.

de Ruiter, L. E., & Bers, M. U. (2021). The Coding Stages Assessment: development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. *Computer Science Education*, 1-30.

Adnan, A., & Romli, R. (2022). A Comparative Evaluation on Methods of Teaching Computer Programming. In *International Conference of Reliable Information and Communication Technology* (pp. 571-582). Springer, Cham.

Tanna, P., Lathigara, A., & Bhatt, N. (2022). Implementation of Problem Based Learning to Solve Real Life Problems. *Journal of Engineering Education Transformations*, *35*(Special Issue 1).

Omeh, C. B., Olelewe, C. J., & Nwangwu, E. C. (2022). Impact of teaching computer programming using innovative pedagogy embedded with live online lectures and related tools: A randomized control trial. *Computer Applications in Engineering Education*.

Hwang, G. J., Tung, L. H., & Fang, J. W. (2022). Promoting Students' Programming Logic and Problem-Solving Awareness With Precision Feedback: A Two-Tier Test-Based Online Programming Training Approach. *Journal of Educational Computing Research*, 07356331221087773.